Final Report • May 1993

A REAL-TIME SPOKEN-LANGUAGE SYSTEM FOR INTERACTIVE PROBLEM SOLVING

Principal Investigators:

Patti J. Price, Director
Speech Research Program

Robert C. Moore, Principal Scientist
Artificial Intelligence Center

SRI Project 8900

Contract N00014-90-C-0085

Prepared for:

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Attn: LCDR Robert D. Powell, USN
OCNR Code 113D

ARPA/SISTO
3701 North Fairfax Drive
Arlington, VA 22203

Attn: Dr. Thomas Crystal

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency of the U.S. Government.

Final Report • May 1993

A REAL-TIME SPOKEN-LANGUAGE SYSTEM FOR INTERACTIVE PROBLEM SOLVING

Principal Investigators:

Patti J. Price, Director
Speech Research Program

Robert C. Moore, Principal Scientist
Artificial Intelligence Center

SRI Project 8900
Contract N00014-90-C-0085

Prepared for:

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Attn: LCDR Robert D. Powell, USN
OCNR Code 113D

and

ARPA/SISTO
3701 North Fairfax Drive
Arlington, VA 22203

Attn: Dr. Thomas Crystal

Approved:

Donald L. Nielson, Vice President
Computing and Engineering Sciences Division

SRI International 333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX (415) 326-5512 • Telex 334486

CONTENTS

1 INTRODUCTION 1

2 SPEECH RECOGNITION 1

2.1 Improving Speech Recognition Accuracy 1
2.2 Improving Speech Recognition Speed 2
2.3 Improving Speech Recognition Robustness 2
2.4 Improving Speech Recognition Portability 3

3 NATURAL-LANGUAGE UNDERSTANDING 3

3.1 The Template Matcher 4
3.2 Gemini 4
3.3 Integration of the Template Matcher and Gemini 5

4 SPEECH AND NATURAL-LANGUAGE INTEGRATION 4

4.1 Dealing with Recognition Errors 6
4.2 Detecting and Correcting Verbal Repairs 8

5 DATA COLLECTION AND ANALYSIS 9

6 PERFORMANCE EVALUATION 10

7 DEMONSTRATION SYSTEMS 11

8 RELATED ACTIVITIES 12

REFERENCES 15

APPENDIX: PAPERS WRITTEN ON THE PROJECT A-1

"SRI International Results February 1992 ATIS Benchmark Test" A-3
"Integrating Multiple Knowledge Sources For Detection and Correction of Repairs in Human-Computer Dialog" A-9
"A System for Labeling Self-Repairs in Speech" A-17
"Prosody, Syntax and Parsing" A-27
"Designing the Human Machine Interface in the ATIS Domain" A-35
"Spontaneous Speech Effects in Large Vocabulary Speech Recognition Applications" A-41
"The DECIPHER Speech Recognition System" A-47
"CSR Corpus Development" A-51
"Gemini: A Natural Language System for Spoken-Language Understanding" A-55
"Focus and Ellipsis in Comparatives and Superlatives: A Case Study" A-63
"Multi-Site Data Collection and Evaluation in Spoken Language Understanding" A-83
"Integrating Two Complementary Approaches to Spoken Language Understanding" A-89
"A Template Matcher for Robust NL Interpretation" A-93

"SRI's Experience with the ATIS Evaluation" A-99
"Efficient Bottom-Up Parsing" A-101
"Speech Recognition in SRI's Resource Management and ATIS Systems" A-105
"Performance of SRI's DECIPHER™ Speech Recognition System on DARPA's CSR Task" A-111
"Reduced Channel Dependence for Speech Recognition" A-117
"Integrating Natural Language Constraints into HMM-based Speech Recognition" A-123
"Training Set Issues in SRI's DECIPHER Speech Recognition System" A-127
"Evaluation of Spoken Language Systems: the ATIS Domain" A-131
"Spoken Language System Integration and Development" A-137
"Subject-Based Evaluation Measures for Interactive Spoken Language Systems" A-141
"The Relationship of Filled-Pause F0 to Prosodic Context" A-147
"Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction" A-157
"User Behaviors Affecting Speech Recognition" A-163

1 Introduction

SRI International (SRI) has carried out a three-year project to develop spoken-language understanding technology for interactive problem solving, featuring real-time performance, large vocabulary, high semantic accuracy, habitability, and robustness. This technology has been developed using an Air Travel Information System (ATIS) as a prototype application. We have developed technology that enables a user to retrieve airline schedules, fares, and related information by means of spoken natural-language queries. We have evaluated this technology in four ATIS benchmark evaluations, and we have incorporated it into a demonstration system, which we have also used for data collection.

This final report consists of a summary of the research and other activities carried out under the project, followed by an Appendix containing 26 technical papers describing work performed on the project. The report covers work on speech recognition, natural-language understanding, speech and natural-language integration, data collection and analysis, performance evaluation, demonstration systems, and related activities.

2 Speech Recognition

SRI's continuous speech, speaker-independent DECIPHER* speech recognizer is based on tied-mixture hidden Markov models. It uses six cepstra- and energy-based features generated from a filterbank computed via fast Fourier transforms and high-pass filtering in the log-spectral-energy domain. Pronunciation variability is modeled through probabilistically pruned linguistic rules. Cross-word acoustic and phonological models are used. Recognizers trained separately on male and female speech are run in parallel, and a backed-off bigram language model is used to reduce perplexity.

SRI's speech recognition effort over the course of the project has involved several tasks: improving speech recognition accuracy, improving speech recognition speed, improving speech recognition robustness, and improving speech recognition portability. Each of these tasks is described briefly below.

2.1 Improving Speech Recognition Accuracy

We have investigated several lines of research that have led to improvements in speech recognition accuracy. Increasing the amount of training usually leads to improved performance, both by providing more training of existing models and by allowing for robust estimation of more detailed models with more parameters. Other techniques

*All product names and trademarks used in this document are the properties of their respective owners.

that have led to improvements include corrective training algorithms, separate modeling of male and female speakers, implementation of tied-mixture ("semi-continuous") hidden Markov models, algorithms for combining and differentially weighting different sources of training data for bigram and class language models, and an improved back-off estimation algorithm (Cohen et al., 1990; Murveit, Weintraub, and Cohen, 1990; Murveit, Butzberger, and Weintraub, 1991; Butzberger et al., 1992; Murveit, Butzberger, and Weintraub, 1992a).

2.2 Improving Speech Recognition Speed

We have achieved significant speed gains in our recognition training algorithms by implementing server-client protocols for hidden-Markov-model training distributed over several machines; these improvements reduced training times by an order of magnitude. In addition, through software engineering, we have achieved high-speed static grammar compilation for higher-order N-gram language models. The new techniques implemented have reduced the size of the grammars, which means that the DECIPHER system runs more quickly and requires less memory. In addition, we have implemented fast-search recognition algorithms for near-real-time recognition of large vocabularies. Finally, we have recoded our front-end to achieve computation speeds about twice faster than real time.

2.3 Improving Speech Recognition Robustness

Our analysis of errors in benchmark tests revealed that much of the discrepancy between ARPA Resource Management task results and ATIS task results was related to spontaneous speech phenomena not observed in the Resource Management data. Therefore, much of our effort on this project has been focused on modeling these spontaneous speech phenomena, including more appropriate word modeling and better models of breath noise and pause fillers (Butzberger et al., 1992; Murveit, Butzberger, and Weintraub, 1992a). Modeling of verbal repairs is reviewed later in Section 4.2 under the topic of integration, since it involves both speech and natural language.

We have also improved DECIPHER's robustness to time-invariant or slow-moving linear channel effects by implementing RASTA filtering (high-pass filtering in the log spectral domain) to improve channel robustness. We have demonstrated the effectiveness of this type of filtering in a set of experiments involving speech passed through a digital filter, two radically different microphones, and digit recognition over the telephone (Murveit, Butzberger, and Weintraub, 1992b).

2.4 Improving Speech Recognition Portability

We have explored several aspects of the issue of portability, including porting to new vocabularies, to new language models, and to new platforms. We have tested these components by porting the ATIS system to the new, 46-city ATIS database, and, in a coordinated internally funded effort, porting to an online version of the system connected to the *Official Airline Guide* Electronic Edition.

We began work in the area of portability by developing mechanisms for the automatic generation of baseforms, applications of rules and the creation of word-models based on existing training data from other domains. We have also developed techniques for porting more easily to new vocabularies by modeling morphophonological correspondences.

Since different tasks and platforms may require different language models, we have written several software tools to manipulate and convert grammars of different formats into a single format for use by the recognition system. A variety of grammars (e.g., back-off N-gram, word-pair, all-word, and finite-state) are supported in the recognizer in a uniform and consistent framework. The mechanism allows for searching parallel recognition paths that contain separate male and female acoustic-phonetic models; it also supports dynamic grammars and N-best search algorithms.

Since porting to different platforms can require changes in the allowable system size, we have explored clustering techniques that allow larger tasks to fit on a smaller machine, and also allow for more detailed models without computational explosion. Since porting to many applications will not allow for additional hardware, we have implemented a system (in conjunction with support from other related projects) that needs only a SUN workstation and an analog-to-digital converter, with no digital-signal-processing board required.

3 Natural-Language Understanding

Under this project, SRI's research on natural-language understanding for spoken-language systems has proceeded along two lines. The shorter-term line of research has focused on the Template Matcher, a module that constructs database queries by searching the user input for key words and phrases characteristic of the most common query types for a given task, ignoring parts of the input that it does not understand. This approach is robust to many kinds of nonstandard use of language, standard language that is simply unanticipated, and speech recognition errors in noncritical parts of the utterance. It is limited, however, in its ability to extract information that depends on structural relationships among words and phrases. A longer-term effort is focused on more sophisticated syntactic and semantic analysis of the input, using a unification-grammar-based natural-language processing system called Gemini. This system is capable of analyzing more complex semantic relationships than the

Template Matcher, but is more fragile, by itself, to unanticipated variation in query phrasing. These two approaches to natural-language understanding are described below, along with our current and planned methods for integrating them.

3.1 The Template Matcher

The Template Matcher (Jackson et al., 1991) operates by filling templates from information it finds in the input utterance. Templates represent skeletal database queries for common types of requests in a particular database query task. For the ATIS task, the topics for which query templates have been defined include flights, fares, ground transportation, the meanings of codes and headings, aircraft, cities, airlines, and airports. Each template has a set of key words and phrases that tend to signal the corresponding type of query and a set of slots that the Template Matcher fills using words and phrases found in the input. For example, for the flight template, the keywords include *flight*, *fly*, and *go*, and the word *from* followed by an airport or city name will cause the "origin" slot to be filled with that name.

For each template, a score is computed that is roughly the percentage of words in the sentence that contribute in some way to matching or filling the template. If the utterance fails to contain any of the keywords that normally signal the template, this basic score is reduced by a factor that varies from template to template. For each input utterance, the Template Matcher tries to fill each kind of template, and the one with the best score is used to construct the database query, provided its score is greater than a certain "cut-off" parameter. The selected filled template is then translated into a database query.

3.2 Gemini

Gemini (Dowding et al., 1993a, 1993b) is a parsing and semantic interpretation system based on unification grammar. This means that grammatical categories incorporate features that can be assigned values, and when grammatical category expressions are matched in the course of parsing or semantic interpretation, these feature assignments are unified; that is, the resulting category expression is the most general expression consistent with all the feature constraints of the expressions being matched.

Processing starts in Gemini when syntactic, semantic, and lexical rules are applied by a bottom-up all-paths "constituent" parser to populate a chart with edges containing syntactic, semantic, and logical form information. Then, a second "utterance" parser is used to apply a second set of syntactic and semantic rules that are required to span the entire utterance. If no semantically acceptable utterance-spanning edges are found during this phase, a component to recognize and correct verbal repairs is applied. When an acceptable interpretation is found, a set of parse preferences is used to choose a single best interpretation from the chart to be used for subsequent

processing. Quantifier scoping rules are applied to this best interpretation to produce a scoped logical form. This logical form is operated on by a set of task-specific rules that map it into a simplified logical form that closely matches the schema of the database. Finally, a module similar to that used to process the output of the Template Matcher translates simplified logical forms into database queries.

In a fair test on the class A and D utterances in the November 1992 ATIS benchmark test set, Gemini was able to find a complete syntactic analysis for 93.1 percent of the utterances and a complete semantic analysis for 86.0 percent of the utterances.

3.3 Integration of the Template Matcher and Gemini

The fact that Gemini attempts a more complete analysis of an utterance than the Template Matcher does suggests that the Template Matcher will succeed more often than Gemini in finding some interpretation for an utterance, but that when Gemini does find an interpretation, it is more likely to be correct than the Template Matcher. Our experiments with ATIS training data have in fact demonstrated this to be the case. To get the benefits of both approaches, the ATIS system we currently use in benchmark evaluations incorporates both Gemini and the Template Matcher, by first attempting to construct a complete analysis of a query using Gemini, and falling back on the Template Matcher if that fails. That way Gemini gets a chance to give an exact analysis of the input before the Template Matcher attempts an approximate one. The approach proved successful in the November 1992 ATIS NL benchmark test (Pallet et al., 1993), where a system based on the Template Matcher alone had a weighted error of 27.6 percent, while the combination of Gemini with the Template Matcher had a weighted error of only 23.6 percent. The difference was even greater for the "class A" (context-independent) subset of queries, where the system incorporating Gemini had a weighted error of only 14.8 percent, compared to 22.2 percent for the Template Matcher alone.

Under the follow-on project, we intend to undertake a more thorough integration of template matching techniques directly into Gemini. To achieve this integration, we will modify the Gemini utterance-level parser to allow it to skip words in the input and assign a corresponding score to the analysis. Since the utterance grammar in Gemini already incorporates rules for semantically combining a sequence of fragments, we expect this will largely subsume the functionality of the Template Matcher with minimal changes to Gemini. Moreover, the performance of the system should be increased, since the general phrase types that can be combined in this way should cover cases that are not covered by the more specific patterns the Template Matcher currently relies on. The Gemini constituent parser (Moore and Dowding, 1991) has been designed in anticipation of this type of processing, incorporating a novel algorithm that finds all complete grammatical phrases bottom up while using limited prediction from context to control creation of spuriously hypothesized phrases

containing unlicensed syntactic gaps.

4 Speech and Natural-Language Integration

Work on integration of speech and language processing under this project has focused on two problems introduced by spontaneous speech that are not present in analysis of fluent textual natural language: (1) coping with errors in the transcription of the speech caused by imperfect recognition, and (2) coping with disfluencies present even in a perfect transcription caused by speakers' verbal repairs in spontaneous speech. Our results in these areas are discussed below.

4.1 Dealing with Recognition Errors

Our research has addressed the problem of understanding natural language containing recognition errors in two ways. For the near term, we have taken advantage of the robustness of the Template Matcher to accommodate recognition errors. Since the Template Matcher can ignore much of the input utterance, recognition errors in these noncritical parts of the utterance typically do not create errors in understanding. The effect of this robustness to recognition errors can be seen in the November 1992 ATIS benchmark tests (Pallet et al., 1993). In the speech recognition test, SRI's DECIPHER recognizer had a sentence error rate of 33.8 percent for the answerable queries, but in the spoken language system (SLS) test, SRI's ATIS system failed to return the correct answer from the database for only 21.6 percent of these utterances. Viewed this way, the (nonweighted) understanding error rate was only 64 percent of the recognition error rate.

Despite this robustness to recognition error, SRI's SLS ATIS system still had a 41 percent higher error rate than the same natural-language understanding system did when tested with error-free transcriptions of utterances. Thus we have also pursued a longer-term line of research to try to use constraints from natural language to reduce the rate of recognition errors with the goal of improving the overall rate of correct understanding.

In experiments reported in 1990 (Murveit and Moore, 1990), we demonstrated the use of a natural-language grammar to reduce the rate of speech recognition errors in the ARPA Resource Management task. In these experiments we were able to reduce the recognition word error rate by 26 percent by including constraints from a natural-language grammar, for sentences falling within the grammar. The base recognition system was a speaker-dependent version of SRI's DECIPHER recognizer using no grammar (perplexity 1000); the natural-language grammar was a syntactic grammar covering 91 percent of the Resource Management corpus; and the test set consisted of 279 sentences covered by the grammar out of 300 sentences divided evenly among three speakers. The integration architecture used was the dynamic grammar

network approach (Moore, Pereira, and Murveit, 1989), in which the natural-language parser incrementally generates a grammar-state-transition network that limits the word sequence hypotheses considered by the recognizer to those permitted by the grammar.

While these experiments were a success, the fact that the possible recognition hypotheses were restricted to those word sequences permitted by the grammar turned out to be a serious limitation. When we turned to the problem of recognizing and understanding spontaneous speech in the ATIS task (in contrast to the read, carefully formed sentences of the Resource Management task), it became apparent that there was very little prospect of writing a grammar that would cover all, or nearly all, of what people would actually say spontaneously to a spoken-language system. A more robust method for applying natural-language constraints in recognition was clearly required. Under this project, we have begun exploring the use of the Gemini system to guide the recognizer to favor more semantically meaningful recognition hypotheses in a way that maintains robustness by making use of information provided by Gemini even when the system fails to obtain a complete semantic analysis.

Our new approach is based on the observation that, even when the grammar fails to find a complete analysis of an utterance, it is usually able to find a small number of phrases that span the utterance. This suggests using the natural-language grammar to compute a language model score for a word sequence hypothesis based on the minimal number of grammatical phrases needed to span the hypothesis. The language model score can be computed as the number of phrases times a parameter optimized to maximize overall performance. The overall scoring formula for recognition hypotheses is then

$$S = R + \alpha G,$$

where R is the score produced by the recognizer (which can incorporate an N-gram statistical language model), G is the grammar score (the minimal number of grammatical phrases needed to span the hypothesis), and α is the parameter to scale the grammar score appropriately to combine with the recognition score. This parameter can be looked on as a "phrase-transition weight" parallel to the "word-transition weight" often used in recognizers to minimize insertion errors.

We have carried out an initial experiment using this model, and the result appears very encouraging. To simplify running this experiment, we used an N-best integration of DECIPHER with SRI's Gemini natural-language processing system. For 100 ATIS training sentences, DECIPHER produced an ordered list of the 20 best-scoring word string hypotheses, using both acoustic models and a bigram language model. Where the top 20 word string hypotheses did not contain the reference string, we added it at the bottom of the list. (This was done to overcome the limitation of the N-best approach that N may have to be very large to avoid pruning errors. Other architectures that we are exploring do not suffer from this problem.) We then scored each hypothesis by the smallest number of phrases needed to cover the hypothesis,

using Gemini's syntactic and semantic rules.

When we compared the scores produced by Gemini for the 1-best hypothesis and for the reference string, we found that in 53 cases Gemini gave them the same score (in 27 of these cases the 1-best hypothesis *was* the reference string), in 44 Gemini gave the reference string a higher score, and in only three cases did Gemini rank an incorrect 1-best hypothesis higher than the reference string. So, in the cases where applying natural-language constraints made a difference, Gemini was almost 15 times more likely to prefer the reference string over an incorrect 1-best hypothesis. Next we looked at what would happen to recognition accuracy if we combined the DECIPHER recognition score with the Gemini language score as we have proposed. We discovered that, for this limited experiment, optimal results were obtained by letting the Gemini score completely dominate the recognition score. That is, optimal results were obtained by limiting consideration to the set of hypotheses given the best score by Gemini, and selecting the hypothesis scored best by DECIPHER from among those. (This would surely not have been the case if significantly more than 20 or 21 hypotheses per utterance had been used.) By doing this, we were able to reduce the total number of recognition errors for the test set from 148 to 116, a reduction of 22 percent, compared with using the DECIPHER recognizer alone.

In comparing these results to our earlier experiments with dynamic grammar networks, it is important to realize that in those experiments we artificially restricted the test set to utterances whose reference transcription could be completely analyzed by the grammar. In the more recent experiments, we made no such restriction, and 25 percent of the test set consisted of utterances for which Gemini could not provide complete analyses at the time the test was performed. So we were, in fact, able to demonstrate the robustness to limitations of the grammar that we were seeking.

4.2 Detecting and Correcting Verbal Repairs

During the past two years, we have investigated the problem of correcting repairs in spontaneous speech. In this type of grammatical disfluency, the speaker intends that the correct interpretation of his or her utterance be gotten by ignoring one or more words or word fragments.

> *How many American airline flights leave Denver on* June *June tenth.*
>
> *Can you give me information on all the flights* from San Fr- no *from Pittsburgh to San Francisco on Monday.*

In this effort, we have developed a notation for describing and annotating repairs (Bear et al., 1993). We have analyzed the repairs occurring in a 10,000 utterance training set of ATIS data, and have developed preliminary methods to recognize and correct repairs combining string matching, acoustic, and natural-language information sources (Bear, Dowding, and Shriberg, 1992; Shriberg, Bear, and Dowding, 1992). In

addition, we have incorporated a component based on those methods into the Gemini system (Dowding et al., 1993a, 1993b).

The mechanism used in Gemini to detect and correct repairs is currently applied as a fallback if no semantically acceptable interpretation is found for the complete utterance. The mechanism finds sequences of identical or related words, possibly separated by a cue word (for example, *oh* or *no*) that might indicate the presence of a repair, and deletes the first occurrence of the matching portion. Since there may be several such sequences of possible repairs in the utterance, the mechanism produces a ranked set of candidate corrected utterances. These candidates are ranked in order of the fewest deleted words. The first candidate that can be given an interpretation is accepted as the intended meaning of the utterance.

The repair correction component currently used in Gemini does not make use of acoustic/prosodic information, but it is clear that acoustics can contribute meaningful cues to repair. In future work, we hope to improve the performance of our repair correction component by incorporating acoustic/prosodic techniques for repair detection developed at SRI (Bear, Dowding, and Shriberg, 1992; Shriberg, Bear, and Dowding, 1992) and elsewhere (Nakatani and Hirschberg, 1993; O'Shaughnessy, 1992).

While it is true that repairs occur relatively rarely in our training data (only three percent of utterances, when simple word fragments are excluded), their rate of occurrence can be expected to increase as speakers become more comfortable talking with a computer. Rates of repair for human-human communication have been reported as high as 34 percent (Levelt '983) for descriptions of visual patterns.

5 Data Collection and Analysis

Early in the project SRI produced a functional equivalent of the data-collection environment for the ATIS task developed by Texas Instruments (TI), and used it to collect and process data from 10 subjects using the TI protocols. We found in experiments based on variations in this system that more constrained scenarios should be used, that familiarization sessions should be used, and that subjects can adapt to small vocabularies (which has important implications for scaling the technologies to various platforms) (Bly et al., 1990).

Through our participation in the MADCOW multi-site ATIS data collection effort (MADCOW, 1992; Hirschman et al., 1993), we have collected training and test data (speech, transcriptions, and logfiles) using SRI's ATIS system, including over 100 speakers, over 200 scenarios, and over 3000 utterances for the 11-city version of the ATIS relational database. We have also collected over 500 utterances in the new 46-city version of the ATIS database. In addition, we have collected data from 16 speakers (32 scenarios, 508 utterances) using two systems: SRI's DECIPHER recognizer hooked up to MIT's TINA NL, and the standard SRI data collection system.

Each speaker solved one scenario using each system, so that user behavior and satisfaction could be compared.

We have carried out extensive analyses of human-machine problem solving using the SRI ATIS system. We have analyzed user satisfaction and system performance as a function of system errors, user experience, and instructions to users, and explored trade-offs of speed vs. accuracy (Shriberg, Wade, and Price, 1992). Our work has shown evidence that, in the face of system word error rates above about 20 percent, users will tend to adapt their speech style (as well as their language) to reduce the error rate (Wade, Shriberg, and Price, 1992).

6 Performance Evaluation

SRI has participated in every ARPA spoken-language benchmark evaluation conducted during the course of the project. Our progress in natural-language understanding, spoken-language understanding, and speech recognition as measured by the ATIS benchmark tests is presented in Table 1. The "NL" results measure natural-language understanding performance in terms of the response error for retrieving the correct answer from the ATIS database, given a correct word-level transcription of the subject's utterance. The "SLS" results measure spoken-language understanding performance starting from the acoustic signal. Both of these are measured in terms of weighted utterance error percentage, according to which a wrong answer is counted as twice as bad as not answering at all. The "SPREC" results measure speech recognition performance in terms of word error percentage. "Class A" refers to the subset of utterances that were judged to be answerable queries whose interpretation did not depend on the context of utterance. "Class A+D" refers to all answerable queries, whether or not context is required for their interpretation. "Class A+D+X" refers to all utterances, whether or not they constitute answerable queries. As can be seen from the table, our performance has steadily improved on all measures over the course of the project.

This project also supported the early stages of SRI's work on the ARPA CSR large-vocabulary speech recognition task. In the first benchmark evaluation on that task, we achieved a 16.6 percent word error rate in the verbalized punctuation test and a 17.1 percent word error rate in the non-verbalized-punctuation test (standard bigram language model, speaker-independent, closed 5000-word vocabulary).

In addition to evaluating systems in the benchmark tests, SRI has played a leading role in defining and supporting technology evaluation within the ARPA community. It was SRI that proposed and promoted the ATIS task as a common task for evaluation of spoken-language understanding systems. Patti Price's role in the MADCOW effort has had a major impact on the functioning of the benchmark evaluations. In addition, she has directed efforts in assessing our current benchmarks and searching for new

Test performed	June 90	Feb 91	Feb 92	Nov 92
NL class A	77.8	31.0	22.9	14.8
NL class A+D			31.1	23.6
SLS class A		41.4	32.1	26.5
SLS class A+D			45.4	33.2
SPREC class A		18.0	7.3	5.2
SPREC class A+D			8.4	5.7
SPREC class A+D+X			11.0	9.1

Table 1: SRI error rates in ATIS benchmark tests.

evaluation procedures, developing with MIT a method for end-to-end evaluation that takes into account the whole interaction (Price et al., 1992). Robert Moore has played a major role on the Corpora and Performance Evaluation Committee (CPEC—the predecessor of MADCOW) and the Principles of Interpretation Committee. He also chaired the original ATIS query classification working group and the ATIS relational database working group. In the latter capacity he redesigned the ATIS relational database schema, and supervised other SRI staff in revising the 11-city ATIS database to conform to the new schema. He also developed the minimal/maximal scoring criterion for controlling the inclusion of irrelevant information in database answers. Finally, George Doddington chaired the CSR corpus committee, supported in part by this project, until he took a leave of absence from SRI to become program manager at ARPA.

7 Demonstration Systems

SRI has been a leader in demonstrating spoken-language understanding technology, and has achieved several firsts in this area. We believe that SRI was the first site to develop and demonstrate an ATIS SLS system; this system had a 350-word vocabulary and was a near-real-time, speaker-dependent system, using a grammar of perplexity 15-90, depending on how close the sentences used were to the 2900 sentence training set. We also developed a graphical user interface for this system, which was first demonstrated in August of 1990. Later we developed a new, X-based interface to the SLS ATIS system, which allowed demonstrations to be given from any machine running X-windows.

Our next step in demonstration system development was to improve accuracy for speaker-independent recognition while maintaining the real-time speed requirement. We believe that we were the first to use our SLS for data collection with no wizard in the loop (May 1991). Our next step was to make the system portable, which

we achieved through algorithm improvement (reduction in size requirements). We implemented this system on a laptop Sun workstation, and believe we were the first to demonstrate our SLS off-site, at Carnegie Mellon University in October 1991.

We made further improvements in the user interface of our ATIS system, including better paraphrasing of system's understanding, easier to read displays, better handling of system error messages, and simpler control of context mechanism. In addition, we added visual interest by including digitized graphics and improved user friendliness by including a tape-recorder like interface to allow the user to move through background material. This interface was integrated with 2 other demonstrations (telephone banking and *Wall Street Journal* dictation) and delivered to ARPA. This last effort was coordinated with the Real-Time Hardware project and internal funding.

8 Related Activities

In addition to the work in support of performance evaluation described in Section 6, SRI has played a major role, with support from this project, in committee and other work ancillary to the administration of the ARPA Spoken Language Program:

- Patti Price, Robert Moore, and George Doddington have served on the Spoken Language Coordinating Committee.
- Patti Price has served on the Standing Committee for planning ARPA Speech and Natural Language (now Human Language Technology) workshops, including chairing this committee from February, 1992, through March, 1993, and developing a set of documented procedures and guidelines for this series of workshops in the form of a "constitution" for the workshops.
- Patti Price has served on numerous workshop planning committees, including those for June 1990, February 1991 (which she chaired), February 1992, and March 1993.

In connection with these activities, the project has supported the participation of SRI staff in the following ARPA-related administrative meetings:

- Patti Price attended an ATIS development meeting at Texas Instruments in Dallas, Texas, just before the 1990 ICASSP meeting. In addition, she attended the ARPA workshop planning meeting in Washington, DC.
- Patti Price attended an Spoken Language Coordinating Committee meeting in July 1990 in Boston, Massachusetts.
- Patti Price and Robert Moore hosted the Spoken Language Coordinating Committee meeting November 1990.

- SRI hosted the Fourth ARPA Workshop on Speech and Natural Language, Asilomar Conference Center, 19–22 February 1991, followed by an open-house with demonstrations of SRI's technology. Patti Price chaired the workshop and edited the proceedings.

- Robert Moore hosted an Spoken Language Coordinating Committee meeting immediately following the Asilomar Workshop.

- Robert Moore represented SRI at the CPEC meeting in Cambridge, Massachusetts, in March 1991 and was joined by Patti Price at the Spoken Language Coordinating Committee meeting 19–20 March 1991.

- Robert Moore and Patti Price attended the Spoken Language Coordinating Committee held at AT&T Bell Laboratories, Murray Hill, New Jersey, in July 1991.

- Doug Appelt, Robert Moore, Hy Murveit, and Patti Price attended the Spoken Language Coordinating Committee meeting in Pittsburgh, Pennsylvania, in October 1991.

- Robert Moore, Hy Murveit, and George Doddington attended the Spoken Language Coordinating Committee meeting at the National Institute of Standards and Technology, Gaithersburg, Maryland, in March, 1992.

- Robert Moore and George Doddington attended the Spoken Language Coordinating Committee meeting in August 1992, at BBN Systems and Technologies, Cambridge, Massachusetts.

- Patti Price attended and chaired a meeting of the Standing Committee on ARPA workshops in Speech and Natural Language, 10 September 1992 at SAIC in Washington, DC.

This project has also supported the participation of SRI staff members in many important technical and professional meetings:

- ARPA-sponsored Speech and Natural Language Workshops in June 1990, February 1991, and February 1992, the ARPA Spoken Language Technology Workshop in January 1993, and the ARPA Human Language Technology Workshop in March 1993 were all attended by several SRI participants under support of this project. The papers presented at these workshops reporting work on the project are listed in the References section of this report.

- Jared Bernstein, Michael Cohen, Hy Murveit, Patti Price and Mitch Weintraub attended the 1990 International Conference on Acoustics, Speech, and Signal

Processing in Albuquerque, New Mexico. partially supported by this project. Two papers reporting work on the project, by Murveit and Moore (1990) and Cohen et al. (1990), were presented.

- Robert Moore and John Bear attended the 28th Annual Meeting of the Association for Computational Linguistics in Pittsburgh, Pennsylvania, in June 1990. John Bear presented a paper at this meeting based on joint work with Patti Price that was partially supported by this project (Bear and Price, 1990).

- A paper was prepared, supported by this project, for presentation at the Kobe ICSLP in November 1990, describing SRI's SLS integration and development (Price et al., 1990).

- John Butzberger attended the International Conference on Acoustics, Speech, and Signal Processing in Toronto, Canada, in May 1991.

- Robert Moore and John Dowding attended the 29th Annual Meeting of the Association for Computational Linguistics in Berkeley, California, in June 1991.

- Patti Price attended the International Congress of Phonetic Sciences in Aix-en-Provence in August 1991. She also made several laboratory visits: University of Eindhoven/Phillips Research Center (a laboratory focused on intonation analysis), Cap Gemini R and D in Paris (a group responsible for system integration and multilanguage porting in the SUNDIAL project of the European ESPRIT program), and CNET-Lannion (a group doing extensive recognition applications, and responsible for dialogue evaluation).

- Hy Murveit attended the IEEE workshop on Speech Recognition at Arden House in December 1991. Dr. Murveit brought a demonstration of SRI's ATIS spoken-language system to Arden House, and spoke in a panel session on Spoken Language Systems describing SRI's efforts and the overall ARPA Spoken Language Program.

- Patti Price served on the Technical Committee for an NSF workshop on Spoken-Language Understanding in February 1992, at which she led a working group on spoken-language understanding.

- Mark Gawron attended the Second Conference on Semantics and Linguistic Theory in Columbus, Ohio, in May 1992, and presented a paper on comparatives.

- John Bear and John Dowding attended the 30th Annual Meeting of the Association for Computational Linguistics in June 1992 at the University of Delaware. They presented a paper, co-authored with Elizabeth Shriberg, on verbal repairs.

- Patti Price attended and delivered an invited talk at a workshop on "Integrating Speech and Natural Language," sponsored by the European Network of Excellence in Language and Speech (ELSNET) and the European Speech Communication Association (ESCA), held in Dublin, July 15-17, 1992.

References

Appelt, D. E., and E. Jackson (1992) "SRI International Results February 1992 ATIS Benchmark Test," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 95–100 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Bear, J., J. Dowding, and E. Shriberg (1992) "Integrating Multiple Knowledge Sources for Detection and Correction of Repairs in Human-Computer Dialog," in *Proceedings 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware, pp. 56–63.

Bear, J., J. Dowding., E. Shriberg, and P. Price (1993) "A System for Labeling Self-Repairs in Speech," Technical Note 522, SRI International, Menlo Park, California.

Bear, J. and P. Price (1990) "Prosody, Syntax and Parsing," in *Proceedings 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, pp. 17–22.

Bly, B., P. J. Price, S. Park, S. Tepper, E. Jackson and V. Abrash (1990) "Designing the Human Machine Interface in the ATIS Domain," in *Proceedings Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania, pp. 136–140 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Butzberger, J., H. Murveit, E. Shriberg, P. Price (1992) "Spontaneous Speech Effects in Large Vocabulary Speech Recognition Applications," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 339–343 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Cohen, M., H. Murveit, J. Bernstein, P. Price, and M. Weintraub (1990) "The DECIPHER Speech Recognition System," in *Proceedings 1990 International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico, pp. 77–80.

Doddington, G. R. (1992) "CSR Corpus Development," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 363–366 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Dowding, J., J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran (1993a) "Gemini: A Natural Language System for Spoken-Language Understanding," to appear in *Proceedings ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey.

Dowding, J., J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran (1993b) "Gemini: A Natural Language System for Spoken-Language Understanding," to appear in *Proceedings 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio.

Gawron, J. M. (1992) "Focus and Ellipsis in Comparatives and Superlatives: A Case Study," in *Proceedings of the Second Conference on Semantics and Linguistic Theory*, C. Barker and D. Dowty, eds., Department of Linguistics, The Ohio State University, Columbus, Ohio.

Hirschman, L., M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallet, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann (1993) "Multi-Site Data Collection and Evaluation in Spoken Language Understanding," to appear in *Proceedings ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey.

Jackson, E. (1992) "Integrating Two Complementary Approaches to Spoken Language Understanding," in *Proceedings ICSLP 92, 1992 International Conference on Spoken Language Processing*, Banff, Alberta, Canada, pp. 333–336.

Jackson, E., D. Appelt, J. Bear, R. Moore, and A. Podlozny (1991) "A Template Matcher for Robust NL Interpretation," in *Proceedings Speech and Natural Language Workshop*, Pacific Grove, California, pp. 190–194 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Levelt, W. (1983) "Monitoring and Self-Repair in Speech," *Cognition*, Vol. 14, pp. 41–104.

MADCOW [Hirschman et al.] (1992) "Multi-Site Data Collection for a Spoken Language Corpus," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 7–14 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Moore, R., D. Appelt, J. Bear, M. Dalrymple, and D. Moran "SRI's Experience with the ATIS Evaluation," in *Proceedings Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania, pp. 147–148 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Moore, R., and J. Dowding (1991) "Efficient Bottom-Up Parsing," in *Proceedings Speech and Natural Language Workshop*, Pacific Grove, California, pp. 200–203 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Moore, R., F. Pereira, and H. Murveit (1989) "Integrating Speech and Natural-Language Processing," in *Proceedings Speech and Natural Language Workshop*, Philadelphia, Pennsylvania, pp. 243–247 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Murveit, H., J. Butzberger, and M. Weintraub (1991) "Speech Recognition in SRI's Resource Management and ATIS Systems," in *Proceedings Speech and Natural Language Workshop*, Pacific Grove, California, pp. 94–100 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Murveit, H., J. Butzberger, and M. Weintraub (1992a) "Performance of SRI's DECIPHER™ Speech Recognition System on DARPA's CSR Task," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 410–414 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Murveit, H., J. Butzberger, and M. Weintraub (1992b) "Reduced Channel Dependence for Speech Recognition," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 280–284 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Murveit, H. and R. Moore (1990) "Integrating Natural Language Constraints into HMM-based Speech Recognition," in *Proceedings 1990 International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico, pp. 573–576.

Murveit, H., M. Weintraub, and M. Cohen (1990) "Training Set Issues in SRI's DECIPHER Speech Recognition System," in *Proceedings Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania, pp. 337–340 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Nakatani, C., and J. Hirschberg (1993) "A Speech-First Model for Repair Detection and Correction", to appear in *Proceedings ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey.

O'Shaughnessy, D. (1992) "Analysis of False Starts in Spontaneous Speech", in *Proceedings ICSLP 92, 1992 International Conference on Spoken Language Processing*, Banff, Alberta, Canada, pp. 931–934.

Pallet, D. S., J. G. Fiscus, W. M. Fisher, and J. S. Garofolo (1993) "Benchmark Tests for the DARPA Spoken Language Program," to appear in *Proceedings ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey.

Price, P. (1990) "Evaluation of Spoken Language Systems: the ATIS Domain," in *Proceedings Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania, pp. 91–95 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Price, P., V. Abrash, D. Appelt, J. Bear, J. Bernstein, B. Bly, J. Butzberger, M. Cohen, E. Jackson, R. Moore, D. Moran, H. Murveit, and M. Weintraub (1990) "Spoken Language System Integration and Development," in *Proceedings ICSLP 90, 1990 International Conference on Spoken Language Processing*, Kobe, Japan, pp. 729–732.

Price, P., L. Hirschman, E. Shriberg, E. Wade (1992) "Subject-Based Evaluation Measures for Interactive Spoken Language Systems," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 34–39 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Shriberg, E., J. Bear, and J. Dowding (1992) "Automatic Detection and Correction of Repairs in Human-Computer Dialog," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 419–424 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Shriberg, E. E., and R. J. Lickley (1992) "The Relationship of Filled-Pause F0 to Prosodic Context," in *Proceedings of the IRCS Workshop on Prosody in Natural Speech*, Technical Report IRCS–92–37, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, Pennsylvania, pp. 201–209.

Shriberg, E., E. Wade, and P. Price (1992) "Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction," in *Proceedings Speech and Natural Language Workshop*, Harriman, New York, pp. 49–54 (Morgan Kaufman Publishers, Inc., San Mateo, California).

Wade, E., E. Shriberg, and P. Price (1992) "User Behaviors Affecting Speech Recognition," in *Proceedings ICSLP 92, 1992 International Conference on Spoken Language Processing*, Banff, Alberta, Canada, pp. 995–998.

Appendix: Papers Written on the Project

This appendix consists of 26 technical papers that describe work performed on the project. Full bibliographic details for these papers may be found in the References section of this report.

SRI INTERNATIONAL RESULTS FEBRUARY 1992 ATIS BENCHMARK TEST

Douglas E. Appelt, Eric Jackson

SRI International
Menlo Park, CA 94025

ABSTRACT

We describe the results that SRI International achieved on the February 1992 ATIS Speech and Natural Language System Test. The basic architecture of the system is described, including a set of parameters capable of altering the system's behavior and processing strategy. We report on several experiments that were run on the February test set to evaluate several processing strategies for both natural-language only and full spoken-language system tests.

1. INTRODUCTION

This paper reports on the results of running SRI International's spoken-language system on the DARPA-sponsored February 1992 test. The system's natural-language processing has been parameterized in several ways to achieve different behaviors. In addition to running our system with what we believed at the time of the test to be the optimal parameter settings to produce our official results, we have conducted some experiments by running the system with a variety of parameter settings. The results of these experiments shed some light on the trade-offs among various SLS and natural-language processing strategies, and provide some interesting data for evaluating the evaluation methodology itself.

2. SYSTEM DESCRIPTION

The SLS system used for the February evaluation is an integration of the SRI DECIPHER speech recognition [1,4,5] system with the SRI TRAVELOGUE natural-language processing system. The integration between these two systems is currently accomplished by a simple serial interface: the best accoustic hypothesis is processed by the NL system to produce the answer to the query.

The DECIPHER System

DECIPHER is a speaker-independent continuous-speech speech recognition system based on tied-mixture Hidden Markov Model (HMM) models. It uses six features, three being vectors (cepstra, delta-cepstra, and delta-delta-cepstra) and three scalars (energy, delta-energy, and delta-delta-energy). These features are computed from a filter bank that is derived via an FFT and high-pass filtered (RASTA filtered) in the log-spectral-energy domain. DECIPHER models pronunciation variability through word networks generated by linguistic rules then pruned probabilistically. There are cross-word acoustic and phonological models. Parallel recognizers were implemented and trained separately on male and female speech. The DECIPHER-ATIS system uses a backed-off bigram language model to reduce the perplexity of the input speech.

The acoustic models were trained on all available ATIS spontaneous and read data (excluding 809 sentences used for system development that include 362 October 1991 dry run sentences and 447 MADCOW sentences). The backed-off bigram language model was trained on the available ATIS spontaneous speech data. This included 14,779 sentences (approximately 150,000 words). The recognition lexicon consisted of all words spoken in all available spontaneous ATIS data. There are also lexical entries for breaths and silence. No catch-all rejection model was used for out-of-vocabulary items. The vocabulary size is 1385 words.

The TRAVELOGUE System

The TRAVELOGUE system consists of a template-matching sentence-analysis mechanism [3] coupled with a context-handling mechanism and a database query generation component.

The template matcher operates by producing templates from the input sentence which then get translated into database queries. The two main components of a template are the template type, which generally corresponds to a relation in the underlying database, and a set of filled slots, which represent constraints present in the query. A template for the sentence "Show me the nonstop flights from Boston" might be of the type "flight" and have an origin slot filled with "Boston" and a stops slot filled with "0." In addition to these components, a template contains an illocutionary force marker (e.g., "show," "how many," "yes/no"), and a list of explicitly requested fields from the relation associated with the

template type. There are 20 different template types and 110 distinct slots.

The template matcher determines the type of template by looking for certain key nouns or key phrases in the sentence. It incorporates a simple noun phrase grammar that allows it to identify phrases containing key nouns. The presence of a key noun in certain contexts (e.g., in a noun phrase preceded by a word like "show") will more strongly trigger the associated template type than an isolated occurrence of that key noun. Conjunctions of noun phrases containing key nouns produce templates with multiple template types.

Slots are filled by matching regular-expression patterns against the input string. For example, "from" followed by an airport or city name may fill the origin slot of the flight template. To find fillers for slots, the template matcher makes use of a lexicon of names and codes, each associated with the appropriate sort, and special grammars for recognizing numbers, dates, and times. For each template type with some key noun or key phrase present in the sentence, the system tries to find the best "slot covering" of the sentence it can. That is, it tries to find the sequence of slot-filling patterns that matches the sentence and consumes as many words as possible. Two constraints are (1) slot filling phrases may not overlap, and (2) no slot may be filled twice with different values. The system incorporates a schematic mapping of the domain, which contains the information as to how entities are related, and allows the system to determine what slots are possible for each template.

In the next stage, the system chooses a single template from the set of candidate templates that have been constructed. It chooses on the basis of several factors, including the type of key that triggered the template and the number of words consumed in filling slots. A template score is then computed for the chosen template, reflecting the proportion of words in the sentence that are considered to be consumed. Words that fill slots or help slots get filled count, as well as function words and certain other words (such as "please") that are ignored for the purposes of scoring. If the template does not score above a threshold, the system chooses not to risk answering the query. The threshold can be varied depending on how much risk of a wrong answer can be tolerated. For evaluation we have found a threshold of about 0.85 to be optimal, while for data collection we use a lower threshold, typically 0.5.

The template matcher incorporates special mechanisms to handle certain types of false starts and complex conjunctions. These phenomena cannot be handled well in a straightforward, unaugmented, template-matching approach.

The template matcher was developed on all the annotated MADCOW data available as of January 1, 1992. In addition, a 3,000-sentence subset of the MADCOW data was annotated with the correct template for each utterance. The template production of the system could be quickly evaluated on these sentences. As of January 1992, the system's performance on this corpus was above 90%.

When a template is produced, the context-handling mechanism of TRAVELOGUE is invoked to determine whether the template for the current sentence should be modified or expanded based on the current state of the dialogue. The system employs a variety of context handling rules, each of which is justified by a plan-based model of dialogue structure similar to that of Grosz and Sidner [2]. The basic model tracks the context of a dialogue by assuming the user is following a plan that involves knowing which database entities satisfy a set of constraints that he or she has in mind when the session commences, because the user has the goal of formulating a travel plan (as opposed to other purposes for which such a database would be useful).

The context mechanism inherits constraints expressed by previous queries in a scenario as long as accumulating these constraints is consistent with knowing a single set of constraints applicable to a single travel plan. Knowing whether this set of constraints is consistent with the overall plan is accomplished by comparing the new slots to a context priority-lattice that establishes a partial order of dependencies among various template slots. Changes in higher-level constraints cause lower-level constraints to be discarded. This general mechanism is supplemented with a mechanism for handling deictic references and references to particular database entities that have appeared in answers to previous questions.

When a template including contextually inherited slots is produced, the TRAVELOGUE produces, optimizes, and runs a PROLOG database query, generating the final answer.

3. OFFICIAL RESULTS

In the February 1992 DARPA ATIS benchmark tests, SRI achieved the following results: In the ATIS speech recognition evaluation, SRI achieved a word recognition error rate of 11.0% and a sentence recognition error rate of 48.7% over all sentences on the test corpus. In the ATIS natural-language-only test, SRI achieved a weighted error rate of 31.1%, with 533 queries answered correctly, 60 incorrectly, and 94 given no answer. In the ATIS spoken-language systems evaluation, SRI achieved

a weighted error rate of 45.4%, with 444 queries answered correctly, 69 incorrectly, and 174 queries given no answer.

We performed an error analysis on the NL-only evaluation results. We examined all the queries that we did not answer or for which we were scored wrong, and tried to ascertain the cause.

Of the sentences that were either incorrect or unanswered, 46% can be attributed to the failure of the template matcher to generate a correct template. Of these failures, 80% could be remedied within the current framework while 20% would require a substantially different approach, such as a parser and grammar that together could provide more structural information about a sentence. We estimate that 12% of the errors were due to the database query generation component, and 18% were due to failures of the context mechanism to identify the correct context. The remaining errors are attributed to the system declining to answer questions when it determined that its uncertainty about the context was too great.

These figures were derived in a highly subjective fashion, but, nevertheless, we feel they give a roughly accurate picture. For a majority of the utterances that caused trouble for the template-generating component, it is clear that adding a new phrase or new slot could solve the problem. The conclusion we draw from this is that a template-matching approach can be highly successful on a domain of about the same complexity as ATIS. How well this type of approach would scale up to a significantly larger domain remains uncertain.

4. ADDITIONAL EXPERIMENTS

We have implemented several parameters that control the behavior of the system. One parameter is the template-matcher score cutoff.

We recognized that if a system failed to respond correctly to a query, it might give incorrect answers to a number of subseqent context-dependent queries, even though the subseqent sentences were processed correctly, given everything the system can determine about the state of the dialogue. Therefore, we have included several parameters that regulate the generation of responses in situations in which, for one reason or another, the state of the context is in doubt.

One such parameter is a cumulative template-score cutoff. We reasoned that if the system answers a series of questions, each of which receives an acceptable, although less than perfect, template score, eventually a point is reached in which the system is so uncertain about the correctness of the accumulated contextual information, that it should, for evaluation purposes, stop answering questions until a query is encountered that definitely sets a new top-level context. This point is detected by multiplying template scores until the cumulative product drops below the level indicated by the cumulative cutoff parameter. Our official results were produced by using values of 0.85 and 0.82 for the template score cutoff and cumulative score cutoff, respectively.

Another parameter controls the choice of one of three possible ways of dealing with the failure to produce an answer for a query. When the system fails to answer a query, it could refuse to answer any further queries until one is found that sets a new top-level context. Although this would be a ridiculous way for a system to behave when interacting with a real user, some preliminary investigation led us to believe that such a strategy was indeed optimal for the evaluation; this is the strategy used to generate our official results. Another possible strategy, which we dub "always answer," is to have the system answer every question in the last previously known context, regardless of how many intermediate queries fail to produce answers. Finally, we have a "usually answer" mode, in which queries are always evaluated in the most recently determined context, unless there is some feature of the query that indicates explicit dependency on a question that was not answered (such as a pronoun or demonstrative reference that could rely on an unanswered query for its resolution).

We ran experiments on our system for the following configurations of parameters on both NL and SLS data. These runs were made by changing only the parameters discussed above, without attempting to influence the behavior of the system in any other way:

1. **Relaxed Cutoff.** We set the template score cutoff to 0.82, and the cumulative cutoff to 0.70. Some of our earlier experiments suggested that these values were optimal for processing speech recognizer output. (Because of an oversight, they were not used in the official test).

2. **Low Scoring Template Strategy.** This strategy sets the template score cutoff and cumulative cutoff to be 0.01. This allows very low scoring templates to be considered as analyses for a sentence. The conservative strategy of not answering questions after failure to produce any template at all until the next context-resetting sentence was still followed.

3. **Maximum Recall Strategy.** This strategy combines the Low-Scoring Template Strategy with the Always Answer strategy. It seeks to maximize recall

by always answering a query whenever any analysis at all is possible. Naturally, precision suffers, because of the increased chance that some of the poorly rated analyses will be wrong.

4. **Maximum Precision Strategy.** We attempted to maximize the system's precision score by setting the template score cutoff and the cumulative cutoff to be 0.99. This strategy causes the system to respond only to templates with perfect scores and to stop answering in context whenever any uncertainty about a template exists. Naturally, because some correct templates will be discarded, recall suffers.

5. **Always Answer Strategy.** The "always answer" context-handling strategy was adopted, keeping the template score cutoff the same as in the official run.

6. **Usually Answer Strategy.** The "usually answer" context-handling strategy was adopted, keeping the template score cutoff the same as in the official run.

5. RESULTS OF EXPERIMENTS

The results we observed for the experiments described in the previous section (as well as our official results on the evaluation) were as follows, ordered by increasing weighted error:

For NL only:

Parameter Settings	Right	Wrong	No Ans	Wtd. Error
Always Answer	554	72	61	29.84
Usually Answer	538	60	89	30.42
Relaxed Cutoff	537	62	88	30.86
Official Results	533	60	94	31.05
Low-Score Template	558	90	39	31.88
Maximum Recall	565	98	24	32.02
Maximum Precision	480	38	169	35.66

For SLS:

Parameter Settings	Right	Wrong	No Ans	Wtd. Error
Always Answer	457	75	155	44.40
Relaxed Cutoff	447	69	171	44.98
Usually Answer	445	69	173	45.27
Official Results	444	69	174	45.40
Low-Score Template	455	86	146	46.29
Maximum Recall	460	93	134	46.58
Maximum Precision	423	62	202	47.45

As can be seen, the predicted parameter settings for Maximum Recall and Maximum Precision did result in the desired recall-precision tradeoff, although neither of these strategies produced the best results as measured by weighted error. It is also interesting to note that, with the exception of the tests for Relaxed Cutoff and Usually Answer configurations (which were in any case very close), the ordering of the results as measured by weighted error was the same for both NL and SLS tests.

6. SLS EVALUATION WITH BBN RECOGNIZER OUTPUT

Because the preliminary results of the February 1992 ATIS benchmark tests suggested that the SRI TRAVELOGUE NL system and the BBN BYBLOS speech-recognition system had both performed particularly well, SRI and BBN collaborated on an experiment to see how well a combined system would have performed on the benchmark test, using the output of BYBLOS as the input to TRAVELOGUE. We took the BYBLOS output from the official February 1992 ATIS SPREC test and ran it through TRAVELOGUE, configured exactly as it was for the official February 1992 ATIS SLS test. So, although this was not submitted as official February 1992 ATIS SLS test output, it is comparable in every respect to the official results obtained by BBN and SRI. The resulting combination produced 482 correct answers, 69 wrong answers, and 136 without answers, for a weighted error of 39.88%.

This experiment may shed some light on the impact of speech-recognition accuracy for SLS performance, if we compare SLS performance with the SRI and BBN recognizers, holding NL processing constant. The improvement of the SLS weighted error from 45.4% to 39.9% represents a error reduction by a factor of 0.12, and was obtained was obtained by running the NL system on input data for which the word error rate on class A and D sentences was improved from 8 4% to 6.2%, an error reduction factor of 0.26. The corresponding sentence error rates were 44.5% and 34.6%, for an error reduction factor of 0.22.

Although the NL processing in TRAVELOGUE is designed to be robust in the face of recognition errors, it is clear that the point of diminishing return on recognition accuracy has not yet been reached, and significant improvements can be obtained if these error rates can be reduced still further.

We did one other experiment with the combination of BYBLOS and TRAVELOGUE, in which we took the BYBLOS SPREC test output and ran it through TRAVELOGUE using the parameter settings that we now believe to be optimal as a result of the experiments reported in the preceding section. This was a combination

of the "always answer" context-handling strategy with the "relaxed cutoff" parameter settings. We felt that this would represent the best performance the system was currently capable of without increasing the basic underlying competence. In this experiment we obtained 495 correct answers, 77 wrong answers, and 117 without answers, for a weighted error of 39.16%.

7. ELIMINATING CLASS X SENTENCES

In addition to the above tests, we ran a test to evaluate the impact of a proposed change to the evaluation procedures to eliminate class-X sentences from the evaluation. Queries are classified as X for a variety of reasons, the most common being that the query lies outside the scope of the database. Although class-X utterances are not counted when computing the scores for NL and SLS evalutions, it may be the case that class-X queries that are clearly outside the scope of the system's processing capabilities could adversely impact the system's ability to track the context, and thus indirectly affect the system's test results.

If the inclusion of class-X sentences in the test were to make a large difference in the scores, it would call into question the success of the effort to eliminate the impact of processing class-X queries from the evaluation results.

To test the impact of class-X sentences on our system, we ran the system configured exactly as it was for the official test, except that all class-X sentences were excluded from consideration. We found that the weighted error decreased by 0.58 for the NL-only test and by 1.0 for the SLS test. While there is an observable "class-X effect," it seems to be relatively small with our system, and would only be noticeable with a processing strategy that based answering decisions on context uncertainty.

8. SUMMARY AND CONCLUSIONS

It is difficult to draw conclusions from these experiments about the efficacy of various parameter settings and processing strategies for improving performance on the evaluation. The results are in fact very similar, and could well be different with a different test set. It is possible to conclude with confidence only that the Maximum Precision strategy is unlikely to yield the lowest weighted error.

The results of these experiments were rather surprising in that we had originally believed that the parameter choices would have a more significant impact on the weighted error than what we observed. Indeed, the results show a surprising insensitivity to parameter choice. It seems to be the case that the weighted error metric disguises differences in system behavior. For example, the Maximum Precision and Maximum Recall strategies produce vastly different behavior on the SLS test: the Maximum Recall strategy answers almost 70 queries to which the Maximum Precision strategy gives no answer. Yet the difference in weighted error for the two strategies is less than one point.

For comparing performance across systems, it is desireable to have a metric for comparing performan across systems that is relatively insensitive to different answering strategies, and therefore has a better chance of truly reflecting the comprehensiveness of a system's coverage of the domain. These experiments demonstrate that the weighted error metric at least comes close to having that property — a fortunate consequence, because it was chosen primarily on the basis of its inuitive appeal. On the other hand, systems with specific characteristics are preferred for particular purposes. For example, when SRI uses its system for MADCOW data collection, it runs in a mode more closely approximating the Maximum Recall strategy, on the theory that producing some answer, even though not perfectly correct, will hold the user's interest and lead to a smoother flowing dialogue than would frequent "I don't understand" responses, even though the experiments indicate that such a strategy is suboptimal for evaluation. These experiments underscore the need to examine multiple properties of a system to arrive at conclusions regarding that system's overall effectiveness at solving user problems, as effectiveness can depend on factors other than the system's ability to obtain a low weighted error.

An important observation is that the five systems with the best scores in the NL evaluation differed by only 3.8 points. We have shown that our system can demonstrate a variation of more than 3 points in weighted error through the selection of different answering strategies holding the basic competence of the system constant. We would therefore be reluctant to conclude that the scores achieved on this benchmark test indicate a clear difference among these five systems in basic competence.

We found it interesting that the Always Answer context strategy would have produced the best results on this evaluation, because this is the most reasonable strategy to employ in a system intended to interact with a user, rather than merely scoring high on the evaluation. If the goal is to evaluate systems under conditions that approximate as much as possible their conditions of use in the real world, it is reassuring that behavior appropriate to the real world would not be inappropriate for the evaluation.

REFERENCES

1. Butzberger, J. et al., "Modeling Spontaneous Speech Effects in Large Vocabulary Speech Applications", Proceedings of the 1992 DARPA Speech and Natural Language Workshop.
2. Grosz, B. and Sidner, C., "Attentions, Intentions, and the Structure of Discourse," *Computational Linguistics*, Vol. 12, No. 3, 1966.
3. Jackson, E. et al., "A Template Matcher for Robust NL Interpretation," Proceedings of the 1991 DARPA Speech and Natural Language Workshop, pp. 190-194.
4. Murveit, H. et al., "Performance of SRI's Decipher Speech Recognition System on DARPA's ATIS Task," Proceedings of the 1992 DARPA Speech and Natural Language Workshop.
5. Murveit, H. et al., "Reduced Channel-Dependence for Speech Recognition," Proceedings of the 1992 DARPA Speech and Natural Language Workshop.

INTEGRATING MULTIPLE KNOWLEDGE SOURCES FOR DETECTION AND CORRECTION OF REPAIRS IN HUMAN-COMPUTER DIALOG*

John Bear, John Dowding, Elizabeth Shriberg†

SRI International
Menlo Park, California 94025

ABSTRACT

We have analyzed 607 sentences of spontaneous human-computer speech data containing repairs, drawn from a total corpus of 10,718 sentences. We present here criteria and techniques for automatically detecting the presence of a repair, its location, and making the appropriate correction. The criteria involve integration of knowledge from several sources: pattern matching, syntactic and semantic analysis, and acoustics.

INTRODUCTION

Spontaneous spoken language often includes speech that is not intended by the speaker to be part of the content of the utterance. This speech must be detected and deleted in order to correctly identify the intended meaning. The broad class of disfluencies encompasses a number of phenomena, including word fragments, interjections, filled pauses, restarts, and repairs. We are analyzing the repairs in a large subset (over ten thousand sentences) of spontaneous speech data collected for the DARPA Spoken Language Program.[1] We have categorized these disfluencies as to type and frequency, and are investigating methods for their automatic detection and correction. Here we report promising results on detection and correction of repairs by combining pattern matching, syntactic and semantic analysis, and acoustics. This paper extends work reported in an earlier paper (Shriberg et al., 1992a).

The problem of disfluent speech for language understanding systems has been noted but has received limited attention. Hindle (1983) attempts to delimit and correct repairs in spontaneous human-human dialog, based on transcripts containing an "edit signal," or external and reliable marker at the "expunction point," or point of interruption. Carbonell and Hayes (1983) briefly describe recovery strategies for broken-off and restarted utterances in textual input. Ward (1991) addresses repairs in spontaneous speech, but does not attempt to identify or correct them. Our approach is most similar to that of Hindle. It differs, however, in that we make no assumption about the existence of an explicit edit signal. As a reliable edit signal has yet to be found, we take it as our problem to find the site of the repair automatically.

It is the case, however, that cues to repair exist over a range of syllables. Research in speech production has shown that repairs tend to be marked prosodically (Levelt and Cutler, 1983) and there is perceptual evidence from work using lowpass-filtered speech that human listeners can detect the occurrence of a repair in the absence of segmental information (Lickley, 1991).

In the sections that follow, we describe in detail our corpus of spontaneous speech data and present an analysis of the repair phenomena observed. In addition, we describe ways in which pattern matching, syntactic and semantic analysis, and acoustic analysis can be helpful in detecting and correcting these repairs. We use pattern matching to determine an initial set of possible repairs; we then apply information from syntactic, semantic, and acoustic analyses to distinguish actual repairs from false positives.

*This research was supported by the Defense Advanced Research Projects Agency under Contract ONR N00014-90-C-0085 with the Office of Naval Research. It was also supported by a Grant, NSF IRI-8905249, from the National Science Foundation. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency of the U.S. Government, or of the National Science Foundation.

†Elizabeth Shriberg is also affiliated with the Department of Psychology at the University of California at Berkeley.

[1]DARPA is the Defense Advanced Research Projects Agency of the United States Government

THE CORPUS

The data we are analyzing were collected as part of DARPA's Spoken Language Systems project. The corpus contains digitized waveforms and transcriptions of a large number of sessions in which subjects made air travel plans using a computer. In the majority of sessions, data were collected in a Wizard of Oz setting, in which subjects were led to believe they were talking to a computer, but in which a human actually interpreted and responded to queries. In a small portion of the sessions, data were collected using SRI's Spoken Language System (Shriberg et al., 1992b), in which no human intervention was involved. Relevant to the current paper is the fact that although the speech was spontaneous, it was somewhat planned (subjects pressed a button to begin speaking to the system) and the transcribers who produced lexical transcriptions of the sessions were instructed to mark words they inferred were verbally deleted by the speaker with special symbols. For further description of the corpus, see MADCOW (1992).

NOTATION

In order to classify these repairs, and to facilitate communication among the authors, it was necessary to develop a notational system that would: (1) be relatively simple, (2) capture sufficient detail, and (3) describe the vast majority of repairs observed. Table 1 shows examples of the notation used, which is described fully in Bear et al. (1992).

The basic aspects of the notation include marking the interruption point, the extent of the repair, and relevant correspondences between words in the region. To mark the site of a repair, corresponding to Hindle's "edit signal" (Hindle, 1983), we use a vertical bar (|). To express the notion that words on one side of the repair correspond to words on the other, we use a combination of a letter plus a numerical index. The letter M indicates that two words match exactly. R indicates that the second of the two words was intended by the speaker to replace the first. The two words must be similar–either of the same lexical category, or morphological variants of the same base form (including contraction pairs like "I/I'd"). Any other word within a repair is notated with X. A hyphen affixed to a symbol indicates a word fragment. In addition, certain cue words, such as "sorry" or "oops" (marked with CR) as well as filled pauses (CF) are also labeled if they occur immediately before the site of a repair.

I	want	fl-		flights	to	boston.		
		M_1-	|	M_1				
what		what	are	the fares				
M_1	|	M_1						
show	me	flights		daily	flights			
		M_1	|	X	M_1			
I	want	a	flight	one	way	flight		
		M_1	|	X	X	M_1		
I	want	to	leave	depart	before	...		
		R_1	|	R_1				
what	are	what	are	the fares				
M_1	M_2	|	M_1	M_2				
... fly	to	boston		from	boston			
	R_1	M_1	|	R_1	M_1			
... fly	from	boston		from	denver			
	M_1	R_1	|	M_1	R_1			
what	are	are	there	any	flights			
X	X	|						

Table 1: Examples of Notation

DISTRIBUTION

Of the 10,000 sentences in our corpus, 607 contained repairs. We found that 10% of sentences longer than nine words contained repairs. In contrast, Levelt (1983) reports a repair rate of 34% for human-human dialog. While the rates in this corpus are lower, they are still high enough to be significant. And, as system developers move toward more closely modeling human-human interaction, the percentage is likely to rise.

Although only 607 sentences contained deletions, some sentences contained more than one, for a total of 646 deletions. Table 2 gives the breakdown of deletions by length, where length is defined as the number of consecutive deleted words or word fragments. Most of the deletions

Deletion Length	Occurrences	Percentage
1	376	59%
2	154	24%
3	52	8%
4	25	4%
5	23	4%
6+	16	3%

Table 2: Distribution of Repairs by Length

Type	Pattern	Freq.
	Length 1 Repairs	
Fragments	$M_1-, R_1-, X-$	61%
Repeats	$M_1 \mid M_1$	16%
Insertions	$M_1 \mid X_1 \ldots X_i M_1$	7%
Replacement	$R_1 \mid R_1$	9%
Other	$X \mid X$	5%
	Length 2 Repairs	
Repeats	$M_1\ M_2 \mid M_1\ M_2$	28%
Replace 2nd	$M_1\ R_1 \mid M_1\ R_1$	27%
Insertions	$M_1 M_2 \mid M_1 X_1 \ldots X_i M_2$	19%
Replace 1st	$R_1\ M_1 \mid R_1\ M_1$	10%
Other	...|...	17%

Table 3: Distribution of Repairs by Type

Match Length	Fill Length 0	1	2	3
1	.82 (39)	.74 (65)	.69 (43)	.28 (39)
2	1.0 (10)	.83 (6)	.73 (11)	.00 (1)
3	1.0 (4)	.80 (5)	1.0 (2)	—
4	1.0 (2)	1.0 (1)	—	—

— indicates *no observations*

Table 4: Fill Length vs. Match Length

were fairly short; deletions of one or two words accounted for 82% of the data. We categorized the length 1 and length 2 repairs according to their transcriptions. The results are summarized in Table 3. For simplicity, in this table we have counted fragments (which always occurred as the second deleted word) as whole words. The overall rate of fragments for the length 2 repairs was 34%.

A major repair type involved matching strings of identical words. More than half (339 out of 436) of the nontrivial repairs (more editing necessary than deleting fragments and filled pauses) in the corpus were of this type. Table 4 shows the distributions of these repairs with respect to two parameters: the length in words of the matched string, and the number of words between the two matched strings. Numbers in parentheses indicate the number of occurrences, and probabilities represent the likelihood that the phrase was actually a repair and not a false positive. Two trends emerge from these data. First, the longer the matched string, the more likely the phrase was a repair. Second, the more words there were intervening between the matched strings, the less likely the phrase was a repair.

SIMPLE PATTERN MATCHING

We analyzed a subset of 607 sentences containing repairs and concluded that certain simple pattern-matching techniques could successfully detect a number of them. The pattern-matching component reported on here looks for identical sequences of words, and simple syntactic anomalies, such as "a the" or "to from."

Of the 406 sentences containing nontrivial repairs, the program successfully found 309. Of these it successfully corrected 177. There were 97 sentences that contained repairs which it did not find. In addition, out of the 10,517 sentence corpus (10,718 – 201 trivial), it incorrectly hypothesized that an additional 191 contained repairs. Thus of 10,517 sentences of varying lengths, it pulled out 500 as possibly containing a repair and missed 97 sentences actually containing a repair. Of the 500 that it proposed as containing a repair, 62% actually did and 38% did not. Of the 62% that had repairs, it made the appropriate correction for 57%.

These numbers show that although pattern matching is useful in identifying possible repairs, it is less successful at making appropriate corrections. This problem stems largely from the overlap of related patterns. Many sentences contain a subsequence of words that match not one but several patterns. For example the phrase "FLIGHT <word> FLIGHT" matches three different patterns:

show the	**flight**		earliest	**flight**	
	M_1	|	X	M_1	
show the	**flight**	time		**flight**	date
	M_1	R_1	|	M_1	R_1

show the	delta	**flight**		united	**flight**
	R_1	M_1	|	R_1	M_1

Each of these sentences is a false positive for the other two patterns. Despite these problems of overlap, pattern matching is useful in reducing the set of candidate sentences to be processed for repairs. Rather than applying detailed and possibly time-intensive analysis techniques to 10,000 sentences, we can increase efficiency by limiting ourselves to the 500 sentences selected by the pattern matcher, which has (at least on one measure) a 75% recall rate. The repair sites hypothesized by the pattern matcher constitute useful input for further processing based on other sources of information.

NATURAL LANGUAGE CONSTRAINTS

Here we describe two sets of experiments to measure the effectiveness of a natural language processing system in distinguishing repairs from false positives. One approach is based on parsing of whole sentences; the other is based on parsing localized word sequences identified as potential repairs. Both of these experiments rely on the pattern matcher to suggest potential repairs.

The syntactic and semantic components of the Gemini natural language processing system are used for both of these experiments. Gemini is an extensive reimplementation of the Core Language Engine (Alshawi et al., 1988). It includes modular syntactic and semantic components, integrated into an efficient all-paths bottom-up parser (Moore and Dowding, 1991). Gemini was trained on a 2,200-sentence subset of the full 10,718-sentence corpus. Since this subset excluded the unanswerable sentences, Gemini's coverage on the full corpus is only an estimated 70% for syntax, and 50% for semantics.[2]

Global Syntax and Semantics

In the first experiment, based on parsing complete sentences, Gemini was tested on a subset of the data that the pattern matcher returned as likely to contain a repair. We excluded all sentences that contained fragments, resulting in a dataset of 335 sentences, of which 179 contained repairs and 176 contained false positives. The approach was as follows: for each sentence, parsing was attempted. If parsing succeeded, the sentence was marked as a false positive. If parsing did not succeed, then pattern matching was used to detect possible repairs, and the edits associated with the repairs were made. Parsing was then reattempted. If parsing succeeded at this point, the sentence was marked as a repair. Otherwise, it was marked as **no opinion**.

Syntax Only		
	Marked as Repair	Marked as False Positive
Repairs	68 (96%)	56 (30%)
False Positives	3 (4%)	131 (70%)

Syntax and Semantics		
	Marked as Repair	Marked as False Positive
Repairs	64 (85%)	23 (20%)
False Positives	11 (15%)	90 (80%)

Table 5: Syntax and Semantics Results

Table 5 shows the results of these experiments. We ran them two ways: once using syntactic constraints alone and again using both syntactic and semantic constraints. As can be seen, Gemini is quite accurate at detecting a repair, although somewhat less accurate at detecting a false positive. Furthermore, in cases where Gemini detected a repair, it produced the intended correction in 62 out of 68 cases for syntax alone, and in 60 out of 64 cases using combined syntax and semantics. In both cases, a large number of sentences (29% for syntax, 50% for semantics) received a **no opinion** evaluation. The **no opinion** cases were evenly split between repairs and false positives in both tests.

The main points to be noted from Table 5 are that with syntax alone, the system is quite accurate in detecting repairs, and with syntax and semantics working together, it is accurate at detecting false positives. However, since the coverage of syntax and semantics will always be lower than

[2]Gemini's syntactic coverage of the 2,200-sentence dataset it was trained on (the set of annotated and answerable MADCOW queries) is approximately 91%, while its semantic coverage is approximately 77%. On a recent fair test, Gemini's syntactic coverage was 87% and semantic coverage was 71%.

the coverage of syntax alone, we cannot compare these rates directly.

Since multiple repairs and false positives can occur in the same sentence, the pattern matching process is constrained to prefer fewer repairs to more repairs, and shorter repairs to longer repairs. This is done to favor an analysis that deletes the fewest words from a sentence. It is often the case that more drastic repairs would result in a syntactically and semantically well-formed sentence, but not the sentence that the speaker intended. For instance, the sentence "show me <flights> daily flights to boston" could be repaired by deleting the words "flights daily," and would then yield a grammatical sentence, but in this case the speaker intended to delete only "flights."

Local Syntax and Semantics

In the second experiment we attempted to improve robustness by applying the parser to small substrings of the sentence. When analyzing long word strings, the parser is more likely to fail due to factors unrelated to the repair. For this experiment, the parser was using both syntax and semantics.

The phrases used for this experiment were the phrases found by the pattern matcher to contain matching strings of length one, with up to three intervening words. This set was selected because, as can be seen from Table 4, it constitutes a large subset of the data (186 such phrases). Furthermore, pattern matching alone contains insufficient information for reliably correcting these sentences.

The relevant substring is taken to be the phrase constituting the matched string plus intervening material plus the immediately preceding word. So far we have used only phrases where the grammatical category of the matched word was either noun or name (proper noun). For this test we specified a list of possible phrase types (NP, VP, PP, N, Name) that count as a successful parse. We intend to run other tests with other grammatical categories, but expect that these other categories could need a different heuristic for deciding which substring to parse, as well as a different set of acceptable phrase types.

Four candidate strings were derived from the original by making the three different possible edits, and also including the original string unchanged. Each of these strings was analyzed by the parser. When the original sequence did not parse, but one of edits resulted in a sequence that parsed, the original sequence was very unlikely to be a false positive (right for 34 of 35 cases). Furthermore, the edit that parsed was chosen to be the repaired string. When more than one of the edited strings parsed, the edit was chosen by preferring them in the following order: (1) $M_1|XM_1$, (2) $R_1M_1|R_1M_1$, (3) $M_1R_1|M_1R_1$. Of the 37 cases of repairs, the correct edit was found in 27 cases, while in 7 more an incorrect edit was found; in 3 cases **no opinion** was registered. While these numbers are quite promising, they may improve even more when information from syntax and semantics is combined with that from acoustics.

ACOUSTICS

A third source of information that can be helpful in detecting repairs is acoustics. In this section we describe first how prosodic information can help in distinguishing repairs from false positives for patterns involving matched words. Second, we report promising results from a preliminary study of cue words such as "no" and "well." And third, we discuss how acoustic information can aid in the detection of word fragments, which occur frequently and which pose difficulty for automatic speech recognition systems.

Acoustic features reported in the following analyses were obtained by listening to the sound files associated with each transcription, and by inspecting waveforms, pitch tracks, and spectrograms produced by the Entropic Waves software package.

Simple Patterns

While acoustics alone cannot tackle the problem of locating repairs, since any prosodic patterns found in repairs are likely to be found in fluent speech, acoustic information can be quite effective when combined with other sources of information, in particular with pattern matching.

In studying the ways in which acoustics might help distinguish repairs from false positives, we began by examining two patterns conducive to acoustic measurement and comparison. First, we focused on patterns in which there was only one matched word, and in which the two occurrences of that word were either adjacent or separated by only one word. Matched words allow for comparison of word duration; proximity helps avoid variability due to global intonation contours not associated with the patterns themselves. We present

here analyses for the $M_1|M_1$ ("flights for <one> one person") and $M_1|XM_1$ ("<flight> earliest flight") repairs, and their associated false positives ("u s air five one one," "a flight on flight number five one one," respectively).

In examining the $M_1|M_1$ repair pattern, we found that the strongest distinguishing cue between the repairs ($N = 20$) and the false positives ($N = 20$) was the interval between the offset of the first word and the onset of the second. False positives had a mean gap of 42 msec ($s.d. = 55.8$) as opposed to 380 msec ($s.d. = 200.4$) for repairs. A second difference found between the two groups was that, in the case of repairs, there was a statistically reliable reduction in duration for the second occurrence of M_1, with a mean difference of 53.4 msec. However because false positives showed no reliable difference for word duration, this was a much less useful predictor than gap duration. F0 of the matched words was not helpful in separating repairs from false positives; both groups showed a highly significant correlation for, and no significant difference between, the mean F0 of the matched words.

A different set of features was found to be useful in distinguishing repairs from false positives for the $M_1|XM_1$ pattern. A set of 12 repairs and 24 false positives was examined; the set of false positives for this analysis included only fluent cases (i.e., it did not include other types of repairs matching the pattern). Despite the small data set, some suggestive trends emerge. For example, for cases in which there was a pause (200 msec or greater) on only one side of the inserted word, the pause was never after the insertion (X) for the repairs, and rarely before the X in the false positives. A second distinguishing characteristic was the peak F0 value of X. For repairs, the inserted word was nearly always higher in F0 than the preceding M_1; for false positives, this increase in F0 was rarely observed. Table 6 shows the results of combining the acoustic constraints just described. As can be seen, such features in combination can be quite helpful in distinguishing repairs from false positives of this pattern. Future work will investigate the use of prosody in distinguishing the $M_1|XM_1$ repair not only from false positives, but also from other possible repairs having this pattern, i.e., $M_1R_1|M_1R_1$ and $R_1M_1|R_1M_1$.

	Pauses after X (only) and F0 of X less than F0 of 1st M_1	Pauses before X (only) and F0 of X greater than F0 of 1st M_1
Repairs	.00	.92
False Positives	.58	.00

Table 6: Combining Acoustic Characteristics of $M_1|XM_1$ Repairs

Cue Words

A second way in which acoustics can be helpful given the output of a pattern matcher is in determining whether or not potential cue words such as "no" are used as an editing expression (Hockett, 1967) as in "...flights <between> <boston> <and> <dallas> <no> between oakland and boston." False positives for these cases are instances in which the cue word functions in some other sense ("I want to leave boston no later than one p m."). Hirshberg and Litman (1987) have shown that cue words that function differently can be distinguished perceptually by listeners on the basis of prosody. Thus, we sought to determine whether acoustic analysis could help in deciding, when such words were present, whether or not they marked the interruption point of a repair.

In a preliminary study of the cue words "no" and "well," we compared 9 examples of these words at the site of a repair to 15 examples of the same words occurring in fluent speech. We found that these groups were quite distinguishable on the basis of simple prosodic features. Table 7 shows the percentage of repairs versus false positives characterized by a clear rise or fall in F0

	F0 rise	F0 fall	Lexical stress	Cont. speech
Repairs	.00	1.00	.00	.00
False Positives	.87	.00	.87	.73

Table 7: Acoustic Characteristics of Cue Words

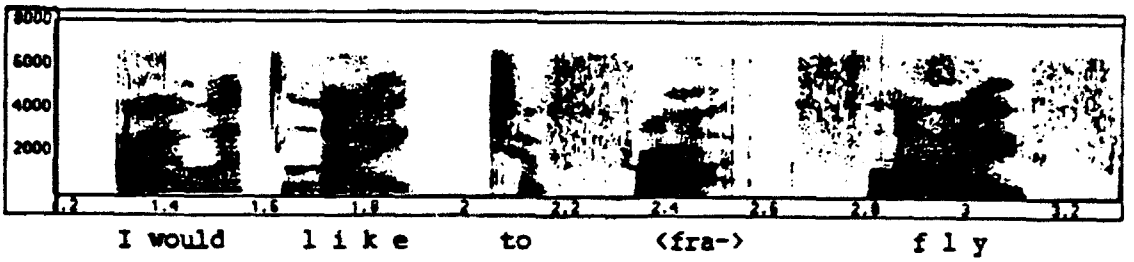


Figure 1: A glottalized fragment

(greater than 15 Hz), lexical stress (determined perceptually), and continuity of the speech immediately preceding and following the editing expression ("continuous" means there was no silent pause on either side of the cue word). As can be seen, at least for this limited data set, cue words marking repairs were quite distinguishable from those same words found in fluent strings on the basis of simple prosodic features.

Fragments

A third way in which acoustic knowledge can assist in detecting and correcting repairs is in the recognition of word fragments. As shown earlier, fragments are exceedingly common; they occurred in 366 of our 607 repairs. Fragments pose difficulty for state-of-the-art recognition systems because most recognizers are constrained to produce strings of actual words, rather than allowing partial words as output. Because so many repairs involve fragments, if fragments are not represented in the recognizer output, then information relevant to the processing of repairs is lost.

We found that often when a fragment had sufficient acoustic energy, one of two recognition errors occurred. Either the fragment was misrecognized as a complete word, or it caused a recognition error on a neighboring word. Therefore if recognizers were able to flag potential word fragments, this information could aid subsequent processing by indicating the higher likelihood that words in the region might require deletion. Fragments can also be useful in the detection of repairs requiring deletion of more than just the fragment. In approximately 40% of the sentences containing fragments in our data, the fragment occurred at the right edge of a longer repair. In a portion of these cases, for example,

"leaving at <seven> <fif-> eight thirty,"

the presence of the fragment is an especially important cue because there is nothing (e.g., no matched words) to cause the pattern matcher to hypothesize the presence of a repair.

We studied 50 fragments drawn at random from our total corpus of 366. The most reliable acoustic cue over the set was the presence of a silence following the fragment. In 49 out of 50 cases, there was a silence of greater than 60 msec; the average silence was 282 msec. Of the 50 fragments, 25 ended in a vowel, 13 contained a vowel and ended in a consonant, and 12 contained no vocalic portion.

It is likely that recognition of fragments of the first type, in which there is abrupt cessation of speech during a vowel, can be aided by looking for heavy glottalization at the end of the fragment. We coded fragments as glottalized if they showed irregular pitch pulses in their associated waveform, spectrogram, and pitch tracks. We found glottalization in 24 of the 25 vowel-final fragments in our data. An example of a glottalized fragment is shown in Figure 1.

Although it is true that glottalization occurs in fluent speech as well, it normally appears on unstressed, low F0 portions of a signal. The 24 glottalized fragments we examined however, were not at the bottom of the speaker's range, and most had considerable energy. Thus when combined with the feature of a following silence of at least 60 msec, glottalization on syllables with sufficient energy and not at the bottom of the speaker's

range, may prove a useful feature in recognizing fragments.

CONCLUSION

In summary, disfluencies occur at high enough rates in human-computer dialog to merit consideration. In contrast to earlier approaches, we have made it our goal to detect and correct repairs automatically, without assuming an explicit edit signal. Without such an edit signal, however, repairs are easily confused both with false positives and with other repairs. Preliminary results show that pattern matching is effective at detecting repairs without excessive overgeneration. Our syntactic/semantic approaches are quite accurate at detecting repairs and correcting them. Acoustics is a third source of information that can be tapped to provide evidence about the existence of a repair.

While none of these knowledge sources by itself is sufficient, we propose that by combining them, and possibly others, we can greatly enhance our ability to detect and correct repairs. As a next step, we intend to explore additional aspects of the syntax and semantics of repairs, analyze further acoustic patterns, and pursue the question of how best to integrate information from these multiple knowledge sources.

ACKNOWLEDGMENTS

We would like to thank Patti Price for her helpful comments on earlier drafts, as well as for her participation in the development of the notational system used. We would also like to thank Robin Lickley for his feedback on the acoustics section, Elizabeth Wade for assistance with the statistics, and Mark Gawron for work on the Gemini grammar.

REFERENCES

1. Alshawi, H, Carter, D., van Eijck, J., Moore, R. C., Moran, D. B., Pereira, F., Pulman, S., and A. Smith (1988) *Research Programme In Natural Language Processing: July 1988 Annual Report*, SRI International Tech Note, Cambridge, England.
2. Bear, J., Dowding, J., Price, P., and E. E. Shriberg (1992) "Labeling Conventions for Notating Grammatical Repairs in Speech," unpublished manuscript, to appear as an SRI Tech Note.
3. Hirschberg, J. and D. Litman (1987) "Now Let's Talk About Now: Identifying Cue Phrases Intonationally," *Proceedings of the ACL*, pp. 163-171.
4. Carbonell, J. and P. Hayes, P., (1983) "Recovery Strategies for Parsing Extragrammatical Language," *American Journal of Computational Linguistics*, Vol. 9, Numbers 3-4, pp. 123-146.
5. Hindle, D. (1983) "Deterministic Parsing of Syntactic Non-fluencies," *Proceedings of the ACL*, pp. 123-128.
6. Hockett, C. (1967) "Where the Tongue Slips, There Slip I," in *To Honor Roman Jakobson: Vol. 2*, The Hague: Mouton.
7. Levelt, W. (1983) "Monitoring and self-repair in speech," *Cognition*, Vol. 14, pp. 41-104.
8. Levelt, W., and A. Cutler (1983) "Prosodic Marking in Speech Repair," *Journal of Semantics*, Vol. 2, pp. 205-217.
9. Lickley, R., R. Shillcock, and E. Bard (1991) "Processing Disfluent Speech: How and when are disfluencies found?" *Proceedings of the Second European Conference on Speech Communication and Technology*, Vol. 3, pp. 1499-1502.
10. MADCOW (1992) "Multi-site Data Collection for a Spoken Language Corpus," *Proceedings of the DARPA Speech and Natural Language Workshop*, February 23-26, 1992.
11. Moore, R. and J. Dowding (1991) "Efficient Bottom-up Parsing," *Proceedings of the DARPA Speech and Natural Language Workshop*, February 19-22, 1991, pp. 200-203.
12. Shriberg, E., Bear, J., and Dowding, J. (1992 a) "Automatic Detection and Correction of Repairs in Human-Computer Dialog" *Proceedings of the DARPA Speech and Natural Language Workshop*, February 23-26, 1992.
13. Shriberg, E., Wade, E., and P. Price (1992 b) "Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction," *Proceedings of the DARPA Speech and Natural Language Workshop*, February 23-26, 1992.
14. Ward, W. (1991) "Evaluation of the CMU ATIS System," *Proceedings of the DARPA Speech and Natural Language Workshop*, February 19-22, 1991, pp. 101-105.

SRI International

Technical Note 522 • February 22, 1993

A System for Labeling Self-Repairs in Speech

Prepared by:

John Bear
Computer Scientist

John Dowding
Computer Scientist

Artificial Intelligence Center
Computing and Engineering Sciences Division

and

Elizabeth Shriberg
Research Linguist

Patti Price
Senior Computer Scientist

Speech Research and Technology Program
Computing and Engineering Sciences Division

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

1. INTRODUCTION

This document outlines a system for labeling self-repairs in spontaneous speech. The system marks the location and extent of a repair, as well as relevant words in the region of the repair. Together these labels determine the relationship between the "error" and the hypothesized "correction." The system is designed to be able to capture distinctions among different repair patterns while remaining easy to learn, apply, and integrate into existing transcription formats. Although the system was originally developed to aid our research on automatic detection and correction of repairs (Shriberg, Bear, & Dowding, 1992; Bear, Dowding & Shriberg, 1992), we hope that it may also prove useful for annotation of spontaneous speech data in related fields.

By "self-repairs" we refer to cases in which one or more words (or word fragments) must be disregarded in determining a speaker's "intended" utterance. Although one can never be sure exactly what a speaker intends, listeners can often reliably make such judgments. For example, given the utterance: "Show me flights from Boston from Denver to Dallas," most listeners would agree that "from Boston" should be disregarded, and that "Show me flights from Denver to Dallas" should be taken as the speaker's intended utterance. Often such judgments can be made on the basis of a transcription alone; listening to the utterance makes available prosodic cues which can greatly facilitate these judgments.

The definition of what constitutes a repair varies in the literature (e.g., Levelt, 1989; Blackmer & Mitton, 1991; Shriberg, Bear & Dowding, 1992). The present system is designed to annotate four types of phenomena:

- repairs involving replacements (as in the example above) or insertions
- repetitions of a string of one or more words ("Show me show me the flight...")
- fresh starts ("Show me the What are the flights...")
- cases involving a word fragment ("Show me the flights from Bos- Denver").

A number of other spontaneous speech phenomena are *not* of concern to this system. For example, filled pauses ("um," "uh") or other fillers ("well," "okay") are not marked unless they occur within an actual repair. This system also does not label silent pauses, uncorrected mispronunciations, repairs involving more than one speaker, and repairs involving a single speaker but in which the correction is a considerable distance (more than one sentence away) from the error.

In Sections 2 through 5, we describe our conventions for marking the site of a repair, and for marking words that distinguish among different repair patterns that we have found useful in our own research. All of the examples included actually occurred in our corpus (our data consisted of human-computer dialog in the air travel planning domain, see MADCOW, 1992). In Section 6, we provide a suggestion for how these labels may be integrated into existing transcription systems.

2. REPAIR SITE

We have adopted a vertical bar (|) notation for marking the site of the repair. The bar marks the resumption of fluent speech; it appears where Hindle (1983) puts his double-dash representing what he calls an "edit signal." In the examples that follow, we place labels on the line below the text.

Example:

List these in increasing in order of increasing fare

|

In the example just cited, the material following the bar ("in order of increasing fare") is a continuation of some of the material that preceded the bar ("List these"). In some repairs, however, the material after the bar constitutes the beginning of a new sentence. These repairs are often referred to as "fresh starts" (e.g., Levelt, 1989).

We mark fresh starts with a special kind of bar notation, so that they can be distinguished from other types of repairs. For fresh starts we use either a period-bar (.|) or a double-bar (||). The .| notation is used for cases in which there is a semantic relationship between the words preceding and following the bar; using this notation commits the labeler to labeling relationships between individual words on either side of the bar (as explained in Section 3). For instance, in the example below, "what is the cheapest" appears on both sides of the bar, and "fare" can be thought of as replacing the word fragment "fl-."

Example:

What is the cheapest fl- what is the cheapest fare

.|

For fresh starts in which a new idea is initiated, we use a double-bar (||) to mark the repair site. Use of the double bar means that the labeler is not committed to marking the relationships between words preceding and following the repair site. In the next example, there is a change in the semantics of the utterance, and although there are matching words on either side of the double-bar (i.e. "does this flight") it would be more difficult to annotate this utterance at the word level because of the presence of many unmatched words.

Example:

What time does this flight arrive where does this flight make a stop

||

Use of the .| versus || notation for repairs that constitute fresh starts is therefore a decision on the part of the labeler that is made by considering both the semantic relatedness of the material preceding and following the repair site, and the degree to which there are word-

by-word correspondences between these two portions of the utterance. A rule of thumb is to use the double-bar for any cases in which it would be difficult to determine word-by-word correspondences.

3. WORD-LEVEL LABELS

Individual words in the region of a repair are annotated with one of four possible labels: *M* (for "matching"), *R* (for "replacement"), *X* (for "insertion" or "deletion") or *C* (for "cue word").

3.1 Matching Words

Repairs often include repetitions of words or phrases. We note these words with the letter *M* (for match) plus a numerical index. Two occurrences of M_i indicate a repetition of the same word.

Examples:

I want to go to | to Boston

M_1 | M_1

I'd like | I'd like to stop in Washington

M_1 M_2 | M_1 M_2

3.2 Replacements

In many cases we want to express the notion of one word replacing another. This we indicate with an *R* and a numerical index.

Examples:

to the city at Atlanta | in Atlanta using ground transportation

R_1 M_1 | R_1 M_1

What are the cheap | cheapest one way flights

R_1 | R_1

In the first example, "in" replaces "at." In both examples the relationship between the two elements constituting the replacement is one of shared grammatical category. In the second example, not only do the two words have the same grammatical category, they are also different morphological forms of the same word.

Finally, in the case of similar but different contractions as illustrated below, we have elected to use both *M* and *R* where appropriate, though clearly there are other reasonable alternatives. To represent the contracted forms, we use a caret (^) to link the associated labels.

Examples:

All right	I'll		I'm	interested in flight five eleven
	M_1^R_1	|	M_1^R_1	

I'd	like		I	would	like	breakfast served
M_1^R_1	M_2	.|	M_1	R_1	M_2	

Note that these examples of contractions differ from the example in Section 3.1. Where the entire contraction is repeated, as in Section 3.1, we simply treat the word as a single unit and annotate it with M_i. When only part of the contraction is repeated, we break the contraction down and annotate each of the parts individually.

3.3 Insertions and Deletions

Words which figure in a repair (typically those which occur between the repair site and a word marked with M or R) and which are not themselves marked with an *M* or *R* are marked with an *X*. *X*s which occur to the left of a vertical bar indicate deletions; those that occur to the right indicate insertions.

Example:

List	the	aircraft		list	types	of	aircraft ...
M_1	X	M_2	.|	M_1	X	X	M_2

This example illustrates a potential difficulty in deciding whether to use X or R. The best we can say here is that there is no obvious syntactic or semantic relationship between "the" and "types of." If we had the same grammatical category repeated, or nouns describing the same semantic class, such as "aircraft/airplanes," then we would use R instead of X.

Since we do not annotate a construction as as a repair unless some of the words were intended to be deleted, we never have an annotation such as " \| X " where nothing to the left of the bar is annotated. We have also never encountered a sentence which we felt ought to be labeled " X \| X ".

3.4 Cues

We label cue words and phrases (such as "I'm sorry") that occur immediately before the repair site with *C*. For cue phrases, each individual word is marked with a C.

Examples:

from	Atlanta	back	to	Pittsburgh	I'm	sorry		back	to	Denver
		M_1	M_2	R_1	C	C	|	M_1	M_2	R_2

to	Atlanta	I	mean	sorry	Dallas	Fort	Worth	to	Atlanta
M_1	C	C	C	|	X	X	X	X	M_1

4. LABELING NONWORDS

4.1 Filled Pauses

We differ from some researchers (e.g. Levelt, 1989; Blackmer & Mitton, 1991) in that we do not label any cases as repairs if simply a filled pause (typically "uh" or "um") is present. We do, however, label filled pauses that occur within a longer repair. These filled pauses are marked with *FP*.

Examples:

Show	me	just	the	economy	class	*fares*	*uh*		*flights*
						R_1	FP	|	R_1

How	long	is	the	layover	in	Denver	uh		in	Dallas
					M_1	R_1	FP	|	M_1	R_1

4.2 Word Fragments

Word fragments occur frequently immediately before a repair site. We indicate fragments by attaching a hyphen to the appropriate label. For example, if we want to indicate that a word is a replacement for a previously uttered word fragment, we add a hyphen to the R_i, as in the following example.

Example:

on	July	fif-		on	July	twentieth
M_1	M_2	R_1-	|	M_1	M_2	R_1

In this example, the labeler's judgment is that "twentieth" is meant to replace the fragment "fif-" which was likely to have been the start of the word "fifteenth."

Previously we have used M_i to indicate repetition of identical words and R_i to indicate two words that are similar but not identical. In cases in which a word fragment like "phila-" is followed by a similar word like "Philadelphia"—that is, in which a labeler feels it is likely that the fragment was the beginning of what would have been a matched word—the label M_i- should be used.

Example:

Also	list	fl-	flights	from	Atlanta	to	Boston...
		M_1-	|	M_1			

Fragments that seem to be neither matched nor replaced by a word to the right of the repair site are labeled with X- .

Show	me	the	s-	flights	that	are	nonstop
			X-	|			

5. REPAIR EXTENT: HOW MUCH TO ANNOTATE

We have been tacitly following some important conventions about how far to the left and right of the repair site words should be labeled. Repairs whose repair site is marked by | or .| follow these conventions: To the left of the vertical bar, we always annotate all of the words to be "deleted" and only those. An X under a word to the left of the bar means it was intended to be "deleted," hence we do not put an X under a word to the left of the bar unless we think it is part of the error. The words to the right of the bar are only labelled if we believe they are part of the "correction." Typically the last word labeled in a correction will be labeled with either an M_i or an R_i, and we do not label the rest of the words in the utterance after that with X.

Example:

	I'd	like	I'd	like	to	stop	in	Washington
Correct:	M_1	M_2	.|	M_1	M_2			
Incorrect:	M_1	M_2	.|	M_1	M_2	X	X	X X

	What	is	the	earliest	flight	leaving	leaving	Boston
Correct:						M_1 |	M_1	
Incorrect:	X	X	X	X	X	M_1 |	M_1	

For fresh starts whose repair site is labeled with ||, we label all words leftward from the repair site to the beginning of the sentence (they should always be either Xs, Cs, or FPs), but do not label any words to the right of the repair site.

Example:

Now could you What is the ground transportation available

X X X ||

6. LABELS IN TRANSCRIPTIONS

For purposes of exposition, we have in this document associated labels with transcriptions simply by placing the labels directly under the words they refer to. In practice, this can be awkward if the utterance is long and/or contains more than one repair, and in general it adds clutter to transcriptions. A simple convention that avoids these problems is to associate an identification number with each repair, and to indicate this number at the repair site in a transcript. The particular sequence of labels associated with the repair can then be listed in a separate file, under the identification number. Because no words are "skipped" when labeling leftward and rightward of the repair site, and since the location of the identification number in the transcript corresponds to the bar in the label sequence, the linking of labels to words in the transcript is completely determined.

Example:

I'd like to f- #001 go at nine #002 ten

001. R_1- | R_1

002. R_1 | R_1

Corrected sentence: I'd like to go at ten.

In the example above, we have used a pound sign (#) followed by a number as an identifier. The format and characters used in identifiers is arbitrary, however; identifiers should be determined individually by researchers to avoid any potential confusion with symbols they use in their own transcription system.

7. ACKNOWLEDGMENTS

This research was supported by the Defense Advanced Research Projects Agency under Contract ONR N00014-90-C-0085 with the Office of Naval Research. It was also supported by a grant, NSF IRI-8905249, from the National Sceince Foundation. The views

and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency of the U.S. Government, or of the National Science Foundation.

8. BIBLIOGRAPHY

Bear, J., Dowding, J. & Shriberg, E. (1992). Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog. *Proceedings of the Association for Computational Linguistics*, pp. 56-63.

Blackmer, E. & Mitton, J. (1991). Theories of monitoring and the timing of repairs in spontaneous speech. *Cognition*, 39, pp. 173-194.

Hindle, D. (1983). Deterministic parsing of syntactic non-fluencies. *Proceedings of the Association for Computational Linguistics*, pp. 123-128.

Levelt, W. (1989). *Speaking: From intention to articulation.* Cambridge, MA: MIT Press.

MADCOW (1992). Multi-site data collection for a spoken language corpus. *Proceedings of the DARPA Speech and Natural Language Workshop*, February 23-26, 1992.

Shriberg, E. E., Bear, J. & Dowding, J. (1992). Automatic detection and correction of repairs in human-computer dialog. *Proceedings of the DARPA Speech and Natural Language Workshop*, February 23-26, 1992.

PROSODY, SYNTAX AND PARSING

John Bear
and
Patti Price
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

April 4, 1990

Abstract

We describe the modification of a grammar to take advantage of prosodic information provided by a speech recognition system. This initial study is limited to the use of relative duration of phonetic segments in the assignment of syntactic structure, specifically in ruling out alternative parses in otherwise ambiguous sentences. Taking advantage of prosodic information in parsing can make a spoken language system more accurate and more efficient, if prosodic-syntactic mismatches, or unlikely matches, can be pruned. We know of no other work that has succeeded in automatically extracting speech information and using it in a parser to rule out extraneous parses.

1 Introduction

Prosodic information can mark lexical stress, identify phrasing breaks, and provide information useful for semantic interpretation. Each of these aspects of prosody can benefit a spoken language system (SLS). In this paper we describe the modification of a grammar to take advantage of prosodic information provided by a speech component. Though prosody includes a variety of acoustic phenomena used for a variety of linguistic effects, we limit this initial study to the use of relative duration of phonetic segments in the assignment of syntactic structure, specifically in ruling out alternative parses in otherwise ambiguous sentences.

It is rare that prosody alone disambiguates otherwise identical phrases. However, it is also rare that any one source of information is the *sole* feature that separates one phrase from all competitors. Taking advantage of prosodic information in parsing can make a spoken language system more accurate and more efficient, if prosodic-syntactic mismatches, or unlikely matches, can be pruned out. Prosodic structure and syntactic structures are not, of course, completely identical. Rhythmic structures and the necessity of breathing influence the prosodic structure, but not the syntactic structure (Gee and Grosjean 1983, Cooper and Paccia-Cooper 1980). Further, there are aspects of syntactic structure that are not typically marked prosodically. Our goal is to show that at least some prosodic information can be automatically extracted and used to improve syntactic analysis. Other studies have pointed to possibilities for deriving syntax from prosody (see e.g., Gee and Grosjean 1983, Briscoe and Boguraev 1984, and Komatsu, Oohira, and Ichikawa 1989) but none to our knowledge have communicated speech information directly to a parser in a spoken language system.

2 Corpus

For our corpus of sentences we selected a subset of a corpus developed previously (see Price *et al.* 1989) for investigating the perceptual role of prosodic information in disambiguating sentences. A set of 35 phonetically ambiguous sentence pairs of differing syntactic structure was recorded by professional FM radio news announcers. By phonetically ambiguous sentences, we mean sentences that consist of the same string of phones, i.e., that suprasegmental rather than segmental information is the basis for the distinction between members of the pairs. Members of the pairs were read in disambiguating contexts on days separated by a period of several weeks to avoid exaggeration of the contrast. In the earlier study listeners viewed the two contexts while hearing one member of the pair, and were asked to select the appropriate context for the sentence. The results showed that listeners can, in general, reliably separate phonetically and syntactically ambiguous sentences on the basis of prosody. The original study investigated seven types of structural ambiguity. The present study used a subset of the sentence pairs which contained prepositional phrase attachment ambiguities, or particle/preposition ambiguities (see Appendix).

If naive listeners can reliably separate phonetically and structurally ambiguous pairs, what is the basis for this separation? In related work on the perception of prosodic information, trained phoneticians labeled the same sentences with an integer between zero and five inclusive between every two words. These numbers, 'prosodic break indices,' encode the degree of prosodic decoupling of neighboring words, the larger the number, the more of a gap or break between the words. We found that we could label such break indices with good agreement within and across labelers. In addition, we found that these indices quite often disambiguated the sentence pairs, as illustrated below.

- Marge 0 would 1 never 2 deal 0 in 2 any 0 guys
- Marge 1 would 0 never 0 deal 3 in 0 any 0 guise

The break indices between 'deal' and 'in' provide a clear indication in this case whether the verb is 'deal-in' or just 'deal.' The larger of the two indices, 3, indicates that in that sentence, 'in' is not tightly coupled with 'deal' and hence is not likely to be a particle.

So far we had established that naive listeners and trained listeners appear to be able to separate such ambiguous sentence pairs on the basis of prosodic information. If we could extract such information automatically perhaps we could make it available to a parser. We found a clue in an effort to assess the phonetic ambiguity of the sentence pairs. We used SRI's DECIPHER speech recognition system, constrained to recognize the correct string of words, to automatically label and time-align the sentences used in the earlier referenced study. The DECIPHER system is particularly well suited to this task because it can model and use very bushy pronunciation networks, accounting for much more detail in pronunciation than other systems. This extra detail makes it better able to time-align the sentences and is a stricter test of phonetic ambiguity. We used the DECIPHER system (Weintraub *et al.* 1989) to label and time-align the speech, and verified that the sentences were, by this measure as well as by the earlier perceptual verification, truly ambiguous phonetically. This meant that the information separating the member of the pairs was not in the segmental information, but in the suprasegmental information: duration, pitch and pausing. As a byproduct of the labeling and time alignment, we noticed that the durations of the phones could be used to separate members of the pairs. This was easy to see in phonetically ambiguous sentence pairs: normally the structure of duration patterns is obscured by intrinsic duration of phones and the contextual effects of neighboring phones. In the phonetically ambiguous pairs, there was no need to account for these

effects in order to see the striking pattern in duration differences. If a human looking at the duration patterns could reliably separate the members of the pairs, there was hope for creating an algorithm to perform the task automatically. This task could not take advantage of such pairs, but would have to face the problem of intrinsic phone duration.

Word break indices were generated automatically by normalizing phone duration according to estimated mean and variance, and combining the average normalized duration factors of the final syllable coda consonants with a pause factor. Let $\bar{d}_i = (d_i - \mu_j)/\sigma_j$ be the normalized duration of the ith phoneme in the coda, where μ_j and σ_j are the mean and standard deviation of duration for phone j. d_p is the duration (in ms) of the pause following the word, if any. A set of word break indices are computed for all the words in a sentence as follows:

$$n = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \bar{d}_i + d_p/70$$

The term $d_p/70$ was actually hard-limited at 4, so as not to give pauses too much weight. The set $\mathcal{A}$ includes all coda consonants, but not the vowel nucleus unless the syllable ends in a vowel. Although the vowel nucleus provides some boundary cues, the lengthening associated with prominence can be confounded with boundary lengthening and the algorithm was slightly more reliable without using vowel nucleus information. These indices n are normalized over the sentence, assuming known sentence boundaries, to range from zero to five (the scale used for the initial perceptual labeling). The correlation coefficient between the hand-labeled break indices and the automatically generated break indices was very good: 0.85.

3 Incorporating Prosody Into A Grammar

Thus far, we have shown that naive and trained listeners can rely on suprasegmental information to separate ambiguous sentences, and we have shown that we can automatically extract information that correlates well with the perceptual labels. It remains to be shown how such information can be used by a parser. In order to do so we modified an already existing, and in fact reasonably large grammar. The parser we use is the Core Language Engine developed at SRI in Cambridge (Alshawi *et al.* 1988).

Much of the modification of the grammar is done automatically. The first thing is to systematically change all the rules of the form $A \rightarrow B\ C$ to be of the form $A \rightarrow B\ Link\ C$, where Link is a new grammatical category, that of the prosodic break indices. Similarly all rules with more than two right hand side elements need to have link nodes interleaved at every juncture: e.g., a rule $A \rightarrow B\ C\ D$ is changed into $A \rightarrow B\ Link_1\ C\ Link_2\ D$.

Next, allowance must be made for empty nodes. It is common practice to have rules of the form $NP \rightarrow \varepsilon$ and $PP \rightarrow \varepsilon$ in order to handle wh-movement and relative clauses. These rules necessitate the incorporation into the modified grammar of a rule $Link \rightarrow \varepsilon$. Otherwise, a sentence such as a wh-question will not parse because an empty node introduced by the grammar will either not be preceded by a link, or not be followed by one.

The introduction of empty links needs to be constrained so as not to introduce spurious parses. If the only place the empty NP or PP etc. could fit into the sentence is at the end, then the only place the empty Link can go is right before it so there is no extra ambiguity introduced. However if an empty wh-phrase could be posited at a place somewhere other than the end of the sentence, then there is ambiguity as to whether it is preceded or followed by the empty link.

For instance, for the sentence, "What did you see _ on Saturday?" the parser would find both of the following possibilities:

- What L did L you L see L empty-NP empty-L on L Saturday?
- What L did L you L see empty-L empty-NP L on L Saturday?

Hence the grammar must be made to automatically rule out half of these possibilities. This can be done by constraining every empty link to be followed immediately by an empty wh-phrase, or a constituent containing an empty wh-phrase on its left branch. It is fairly straightforward to incorporate this into the routine that automatically modifies the grammar. The rule that introduces empty links gives them a feature-value pair: *empty_link=y*. The rules that introduce other empty constituents are modified to add to the constituent the feature-value pair: *trace_on_left_branch=y*. The links zero through five are given the feature-value pair *empty_link=n*. The default value for *trace_on_left_branch* is set to *n* so that all words in the lexicon have that value. Rules of the form $A_0 \rightarrow A_1\ Link_1 \ldots A_n$ are modified to insure that A_0 and A_1 have the same value for the feature *trace_on_left_branch*. Additionally, if $Link_i$ has *empty_link=y* then A_{i+1} must have *trace_on_left_branch=y*. These modifications, incorporated into the grammar-modifying routine, suffice to eliminate the spurious ambiguity.

4 Setting Grammar Parameters

Running the grammar through our procedure, to make the changes mentioned above, results in a grammar that gets the same number of parses for a sentence with links as the old grammar would have produced for the corresponding sentence without links.

In order to make use of the prosodic information we still need to make an additional important change to the grammar: how does the grammar use this information? This area is a vast area of research. The present study shows the feasibility of one particular approach. In this initial endeavor, we made the most conservative changes imaginable after examining the break indices on a set of sentences. We changed the rule $N \rightarrow N\ Link\ PP$ so that the value of the link must be between 0 and 2 inclusive (on a scale of 0-5) for the rule to apply. We made essentially the same change to the rule for the construction verb plus particle, $VP \rightarrow V\ Link\ PP$, except that the value of the link must, in this case, be either 0 or 1.

After setting these two parameters we parsed each of the sentences in our corpus of 14 sentences, and compared the number of parses to the number of parses obtained without benefit of prosodic information. For half of the sentences, i.e., for one member of each of the sentence pairs, the number of parses remained the same. For the other members of the pairs, the number of parses was reduced, in many cases from two parses to one.

The actual sentences and labels are in the appendix. The incorporation of prosody resulted in a reduction of about 25% in the number of parses found, as shown in table 1. Parse times increase about 37%.

In the study by Price *et al.*, the sentences with more major breaks were more reliably identified by the listeners. This is exactly what happens when we put these sentences through our parser too. The large prosodic gap between a noun and a following preposition, or between a verb and a following preposition provides exactly the type of information that our grammar can easily make use of to rule out some readings. Conversely, a small prosodic gap does not provide a reliable way to tell which two constituents combine. This coincides with Steedman's (1989) observation that syntactic units do not tend to bridge major prosodic breaks.

We can construe the large break between two words, for example a verb and a preposition/particle, as indicating that the two do not combine to form a new slightly larger constituent in which they are sisters of each other. We cannot say that no two constituents may combine when they are separated by a large gap, only that the two smallest possible constituents, i.e., the two words, may not combine.

sentence i.d.	*# parses no prosody*	*# parses with prosody*	*parse time no prosody*	*parse time with prosody*
1a	10	4	5.3	5.3
1b	10	10	5.3	7.7
2a	10	7	3.6	4.3
2b	10	10	3.6	4.0
3a	2	1	2.3	2.7
3b	2	2	2.3	3.7
4a	2	1	3.2	4.7
4b	2	2	3.2	5.5
5a	2	1	1.7	2.5
5b	2	2	1.6	2.9
6a	2	1	2.5	2.8
6b	2	2	2.5	4.1
7a	2	1	0.8	1.3
7b	2	2	0.8	1.5
TOTAL	60	46	38.7	53.0

Table 1: The number of parses and parse times (in seconds) with and without the use of prosodic information.

To do the converse with small gaps and larger phrases simply does not work. There are cases where there is a small gap between two phrases that are joined together. For example there can be a small gap between the subject NP of a sentence and the main VP, yet we do not want to say that the two words on either side of the juncture must form a constituent, e.g., the head noun and auxiliary verb.

The fact that parse times increase is due to the way in which prosodic information is incorporated into the text. The parser does a certain amount of work for each word, and the effect of adding break indices to the sentence is essentially to double the number of words that the parser must process. We expect that this overhead will constitute a less significant percentage of the parse time as the input sentences become more complex. We also hope to be able to reduce this overhead with a better understanding of the use of prosodic information and how it interacts with the parsing of spoken language.

5 Corroboration From Other Data

After devising our strategy, changing the grammar and lexicon, running our corpus through the parser, and tabulating our results, we looked at some new data that we had not considered before, to get an idea of how well our methods would carry over. The new corpus we considered is from a recording of a short radio news broadcast. This time the break indices were put into the transcript by hand. There were twenty-two places in the text where our attachment strategy would apply. In eighteen of those, our strategy or a very slight modification of it, would work properly in ruling out some incorrect parses and in not preventing the correct parse from being found. In the remaining

four sentences, there seem to be other factors at work that we hope to be able to incorporate into our system in the future. For instance it has been mentioned in other work that the length of a prosodic phrase, as measured by the number of words or syllables it contains, may affect the location of prosodic boundaries. We are encouraged by the fact that our strategy seems to work well in eighteen out of twenty-two cases on the news broadcast corpus.

6 Conclusion

The sample of sentences used for this study is extremely small, and the principal test set used, the phonetically ambiguous sentences, is not independent of the set used to develop our system. We therefore do not want to make any exaggerated claims in interpreting our results. We believe though, that we have found a promising and novel approach for incorporating prosodic information into a natural language processing system. We have shown that some extremely common cases of syntactic ambiguity can be resolved with prosodic information, and that grammars can be modified to take advantage of prosodic information for improved parsing. We plan to test the algorithm for generating prosodic break indices on a larger set of sentences by more talkers. Changing from speech read by professional speakers to spontaneous speech from a variety of speakers will no doubt require modification of our system along several dimensions. The next steps in this research will include:

- Investigating further the relationship between prosody and syntax, including the different roles of phrase breaks and prominences in marking syntactic structure,
- Improving the prosodic labeling algorithm by incorporating intonation and syntactic/semantic information,
- Incorporating the automatically labeled information in the parser of the SRI Spoken Language System (Moore, Pereira and Murveit 1989),
- Modeling the break indices statistically as a function of syntactic structure,
- Speeding up the parser when using the prosodic information; the expectation is that pruning out syntactic hypotheses that are incompatible with the prosodic pattern observed can both improve accuracy and speed up the parser overall.

7 Acknowledgements

This work was supported in part by National Science Foundation under NSF grant number IRI-8905249. The authors are indebted to the co-Principle Investigators on this project, Mari Ostendorf (Boston University) and Stefanie Shattuck-Hufnagel (MIT) for their roles in defining the prosodic infrastructure on the speech side of the speech and natural language integration. We thank Hy Murveit (SRI) and Colin Wightman (Boston University) for help in generating the phone alignments and duration normalizations, and Bob Moore for helpful comments on a draft. We thank Andrea Levitt and Leah Larkey for their help, many years ago, in developing fully voiced structurally ambiguous sentences without knowing what uses we would put them to.

This work was also supported by the Defense Advanced Research Projects Agency under the Office of Naval Research contract N00014-85-C-0013.

References

[1] H. Alshawi, D. M. Carter, J. van Eijck, R. C. Moore, D. B. Moran, F. C. N. Pereira, S. G. Pulman, and A. G. Smith (1988) *Research Programme In Natural Language Processing: July 1988 Annual Report*, SRI International Tech Note, Cambridge, England.

[2] E. J. Brisco and B. K. Boguraev (1984) "Control Structures and Theories of Interaction in Speech Understanding Systems," COLING 1984, pp. 259-266, Association for Computational Linguistics, Morristown, New Jersey.

[3] W. Cooper and J. Paccia-Cooper (1980) *Syntax and Speech*, Harvard University Press, Cambridge, Massachusetts.

[4] J. P. Gee and F. Grosjean (1983) "Performance Structures: A Psycholinguistic and Linguistic Appraisal," *Cognitive Psychology*, Vol. 15, pp. 411-458.

[5] J. Harrington and A. Johnstone (1987) "The Effects of Word Boundary Ambiguity in Continuous Speech Recognition," *Proc. of XI Int. Cong. Phonetic Sciences*, Tallin, Estonia, Se 45.5.1-4.

[6] A. Komatsu, E. Oohira and A. Ichikawa (1989) "Prosodical Sentence Structure Inference for Natural Conversational Speech Understanding," ICOT Technical Memorandum: TM-0733.

[7] R. Moore, F. Pereira and H. Murveit (1989) "Integrating Speech and Natural-Language Processing," in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 243-247, February 1989.

[8] P. J. Price, M. Ostendorf and C. W. Wightman (1989) "Prosody and Parsing," *Proceedings of the DARPA Workshop on Speech and Natural Language*, Cape Cod, October, 1989.

[9] M. Steedman (1989) "Intonation and Syntax in Spoken Language Systems," Proceedings of the DARPA Workshop on Speech and Natural Language, Cape Cod, October 1989.

[10] M. Weintraub, H. Murveit, M. Cohen, P. Price, J. Bernstein, G. Baldwin and D. Bell (1989) "Linguistic Constraints in Hidden Markov Model Based Speech Recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 699–702, Glasgow, Scotland, May 1989.

8 Appendix

1a. I 1 read 0 a 0 review 2 of 1 nasality 4 in 0 German.

1b. I 0 read 2 a 1 review 1 of 0 nasality 1 in 0 German.

2a. Why 0 are 0 you 2 grinding 0 in 3 the 0 mud.

2b. Why 1 are 0 you 2 grinding 3 in 0 the 1 mud.

3a. Raoul 2 murdered 1 the 0 man 4 with 0 a 1 gun.

3b. Raoul 1 murdered 3 the 0 man 1 with 0 a 0 gun.

4a. The 0 men 1 won 3 over 0 their 0 enemies.

4b. The 0 men 2 won 0 over 1 their 0 enemies.

5a. Marge 1 would 0 never 0 deal 3 in 0 any 0 guise.

5b. Marge 0 would 1 never 2 deal 0 in 2 any 0 guys.

6a. Andrea 1 moved 1 the 0 bottle 3 under 0 the 0 bridge.

6b. Andrea 1 moved 3 the 0 bottle 1 under 0 the 0 bridge.

7a. They 0 may 0 wear 4 down 0 the 0 road.

7b. They 0 may 1 wear 0 down 2 the 0 road.

DARPA Workshop:
Speech & Natural Language
June 1990

Designing the Human Machine Interface in the ATIS Domain

B. Bly P. J. Price S. Park S. Tepper E. Jackson V. Abrash

SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025

Abstract

Spoken language systems for the near future will not handle all of English, but, rather, will be limited to a domain-specific sub-language. Accurate modeling of the sub-language will depend on analysis of domain-specific data. Since no spoken language systems currently have a wide range of users, and since variability across users is expected to be large, we are simulating applications in which a large population of potential users can be sampled. The data resulting from the simulations can be used for system development and for system evaluation. The application discussed here is the air travel domain using the Official Airline Guide (OAG) reformatted in a relational structure.

This study assesses the effects of changes in the simulations on the speech and language of the experimental subjects. These results are relevant to both the experimental conditions for data collection and the design of the human interface for spoken language systems. We report here on five experiments: (1) the effect of longer instructions with examples vs. shorter instructions, using our earlier data collection system, (2) a baseline experiment using a functional equivalent of the data collection effort at Texas Instruments (TI), (3) the use of a more specific version of the scenario used in the baseline experiment, (4) the use of a short, simple familiarization scenario before the main scenario, and (5) in addition to the short familiarization scenario, the use of a finite vocabulary with rejection of sentences with extra-lexical items.

Introduction

The data reported here are part of an endeavor whose goal is to design an appropriate human-machine interface by examining various parameters in a simulated interaction involving air travel planning. The design of the system is such that either a spoken language system (SLS) or a simulation of one can be inserted between the user and the relational database version of the Official Airline Guide data for North American flights and fares. In this way we can gather data for development and evaluation of both the SLS and the user interface.

Perhaps the greatest source of variability in the system is that across subjects. Individuals differ greatly in their language skills, in their problem solving skills, and in their attention spans. It is therefore important to sample a variety of subjects from the relevant population. Individuals are also very adaptable. In many cases, it may be easier to rely on subject adaptability than to try to find technological solutions. However, the dimensions along which humans might adapt are largely unknown for spoken language interfaces. Thus, the simulations provide us with a mechanism to test experimentally various interface strategies that may be appropriate for SLS technology as it develops.

We describe here five experiments aimed at answering various questions about the interface. Our first experiment, the only one reported here that was not based on a functional equivalent of the TI data collection system, investigated the effect of a long set of instructions with examples compared to a shorter set with no examples. The goal of this study was to investigate how much one "poisons the data" by using such examples. The next four experiments were based on either a functional equivalent of the TI system, or a minor variation:

- To serve as a baseline experiment to compare our results to those of TI, and to serve as a control for the other experiments, we collected data in a fashion that imitated the TI system as much as possible.
- To investigate the effects on yield that might result when subjects interpret what a vague scenario might mean, we modified the scenario to fill in details that were unspecified in the original.
- To investigate the first session effect, which was large in our earlier work, we used a simple, short (about 5-minute) familiarization scenario.
- To investigate how well subjects might adapt to a fixed vocabulary, we used a short familiarization scenario, gave subjects a list of about 1000 words, and gave error messages for utterances with words not on that list.

Data Collection Conditions

Except for the first experiment, which was carried out before the functional equivalent of the TI data collection

system had been completed, our aim was to imitate as well as we could the system used by TI for data collection. In particular, we have used the same data from OAG formatted in the same relational structure; the same tool for the "wizard" (NLParse) and accompanying NLParse grammar; the same relational database (Oracle) and interface to NLParse; the same set of tools for communication among subject, wizard, and transcriber; the same subject and experimenter instructions; and the same formatting of tables and other objects displayed on the screens (controlled by Oracle). We used only one of TI's scenarios, planning a family reunion involving family members of various types.

Our data collection differed from that of TI in a few ways that we felt were either unavoidable or unimportant for the resulting data. We are aware of the following differences: our A/D system uses a NEXT machine; our push-to-talk mechanism writes out a time stamp for push and for release (this allows us to calculate the time spent speaking, waiting for an answer and thinking before making the next query, which the TI system does not allow); instead of the color coding used by TI, we use a "ready" prompt when the system is ready to accept speech, a "listening" prompt when the subject is pushing the mouse button, and a "processing" prompt after the subject releases the button and before the answer is sent. We offered a free "DECIPHER" T-shirt to participants in an experimental session.

Data Analysis

Each session was timed from beginning to end, the training scenarios were timed, and the delay until the subject initiated the first utterance was timed. The numbers of words and utterances produced per session were counted, as were the numbers of words and utterances produced during the training scenario. A time stamp was automatically recorded each time the subject used the push-to-talk button, each time a transcription was sent, and each time a response was sent to the subject's screen. This allowed us to determine the average time the subject took after receiving an answer and before formulating a query (thinking time), the average time the subject held down the push-to-talk button (speaking time), and the average time it took the wizard and the wizard's assistant to send the transcription and database response to the subject's screen (subject waiting time). The average number of words per utterance, the average vocabulary size per subject, and the number of sentences outside the restricted vocabulary used in the Fixed Vocabulary Condition were counted. We also counted the number of cancellations subjects used per session, and the number of error messages sent. After the session, all subjects filled out an eleven-item questionnaire designed to assess their subjective impressions of the system and their satisfaction with their interaction with the system. Analyses of these measures were completed for the ten subjects in each of the four conditions that were based on the TI data collection system.

For the word counts, we used the .nli files (see [2]), and used functions to reformat the data so that, for example "845" would count as three words rather than one. Other, similar changes were made to regularize the spellings.

Condition 0: Long Instructions

This condition is the only one that is not based on the TI data collection system; it is based on the system described in [1]. We describe it briefly here since the results were part of the motivation for the two training conditions described below.

This experiment tested the effect of subject instructions on the language produced by the subjects. Two sets of instructions were used: one that included ten grammatical and parsable utterances as examples, and one that included no examples. In all other respects they were identical. Based on previous work, we expected a large effect of experience with the system, so subjects were asked to perform two tasks, and performance was compared across the two tasks as well as between the two sets of instructions. 208z We found a strong interaction between the type of instructions given and the amount of experience the subject had with the system; that is, on a subject's first task, those who received long instructions behaved like the more experienced, second-task subjects on the measures used in the previous study. They also used more complete sentences and did not show the pattern of short, choppy, telegraphic speech demonstrated by the subjects who received a short set of instructions. It is possible, then, to affect the speech the subject addresses to an SLS by providing examples. It is important to note that the effects of longer instructions and additional experience with the system were not additive: new users appear to need either detailed instructions or additional practice time but not both.

The data collected in this experiment was different in important ways from data collected and reported by TI. The sentences, especially those produced by subjects not given examples, were shorter (an average number of 7.4 words per utterance compared to about 12 for the TI data). However, due to the many differences between this interface and that used by TI, it was impossible to reliably attribute these differences to any specific causes. We therefore designed a series of minor modifications of the TI version, as described below.

Condition 1: TI Equivalent

The goal of the "TI" Condition was to establish that our data collection system was a functional equivalent of the TI system, and then to serve as a baseline for the subsequent conditions. We tried to conform as closely as possible to TI's methods, physical setup and materials. In this condition, subjects were read a set of instructions identical to the instructions used by TI, the task they were asked to perform was one of the TI scenarios, and

	TI	SRI-TI
No. utterances	26.2	23.5
No. words	305	298
Words/utterance	11.6	12.7
No. unique words/subj.	83	81
No. unique words/cond.	286	296
Time between utterances	90 sec.	89 sec.

Table 1: SRI-TI Condition Compared with TI Data

the wizard was familiar with NLParse and had practiced, using the transcription and query data released by TI.

The data from our TI Condition seems to match TI's released data very well. As shown in Table 1, the various measures made are all very similar.

Perhaps the most striking difference between TI's data and SRI's in the TI Condition appeared in an analysis of word frequency. We were astonished that the frequencies were so different for "show" (75 occurrences in TI's data vs. 8 in ours). Similar discrepancies showed up for the words "me", "nonstop" and "flights". We then realized that the sentence used by TI as an example demonstrating the use of the mouse and the formatting of the tables, "Show me all the nonstop flights from Atlanta to Philadelphia", had a profound effect on the resulting data (though, of course, these utterances from each speaker were not used in the analysis). In our data collection, we asked the subject to read the first sentence of the scenario while we verified the recording procedure and demonstrated the push-to-talk button.

Condition 2: Task Specificity

We found, in examining both data released by TI and our own data in the TI Condition, that it was often hard to tell how a subject had interpreted a given task, and even which task was being performed. The data could be more valuable if we could ascertain whether and how well the subject completed the task. We also thought that subjects would be more cooperative and the task would be more realistic if they were concentrating on solving the task rather than on exploring the limits of the system. In addition, we suspected that some time might be wasted while the subject tries to figure out what the task is.

To eliminate the effect of individual interpretation of the task and to standardize the task across all subjects, we ran a "Specific Task" Condition. In this condition, subjects were given the same instructions as in our TI Condition. The task they were asked to perform, however, while structurally the same as the tasks performed by TI's subjects and by our own subjects in the TI Condition, was more specific. Instead of leaving the interpretation of certain aspects of the task to the subjects (for instance, find a flight for a person with an "adventurous" lifestyle), we set explicit constraints (find an airplane that holds the fewest number of passengers). In addition, instead of choosing any cities from the database to complete the task, subjects were assigned the origin and destination cities. Each of the ten subjects in this condition used a different set of four cities, determined randomly from the set of cities in the database. In all other aspects, this condition was identical to the previous condition.

We found no significant differences on any of our measures between the subjects in our TI Condition and our Task Specificity Condition. It may be that any benefits gained by subjects not being required to fill in the details themselves were offset by the fact that assigning random cities does not work as well as when subjects pick the cities themselves. For example, several of our subjects had difficulties because they did not realize that Dallas and Fort Worth shared an airport. Subjectively, however, it did appear that subjects completed the assigned task, whereas in the TI Condition, many subjects gave up or quit before fulfilling the various parts of the task required by the scenario. We are working to develop objective measures of this subjective impression of the "dialogue" quality of the collected utterances.

Condition 3: Familiarization

Our past data collection efforts showed a large effect of user experience in human-human interactions and in experimental human-machine interactions [1]. In both conditions, the more domain-experienced speakers produced fewer words, fewer false starts and fewer filler words than did the less-experienced speakers. In addition, subjects elicited fewer error messages in their second scenarios compared to their first. Further, the dramatic effect of one sentence read by all subjects at TI shows just how adaptable subjects can be, at least in an initial session.

In the "Familiarization Condition", after reading the same instructions as in the other conditions, the experimenter stayed in the room with the subject and answered any questions the subject had in finding a single one-way flight between San Francisco and Dallas. The experimenter responded to questions including those regarding the kind of requests the system could handle, the kind of information in the database, and the push-to-talk button. The experimenter also provided possible explanations for any error messages the subject received during the training scenario. The familiarization scenario remained constant across all subjects, although the scenarios that constituted the main task varied among subjects as described in the Task Specificity Condition above. The average length of a training scenario was 6.57 minutes.

Among the various conditions we ran, the largest effect by far was that of the familiarization scenario. As shown in Table 2, subjects who used familiarization sce-

	No Familiarization	With Familiarization
Task time	40 min.	23 min.
Utterances/Task	24	17
Words/Task	276	146
Words/Utterance	12.2	8.7
Format queries	25%	13%
Errors	3.9	1.2
Cancellations	3.8	1.6
Thinking time	46 sec.	34 sec.
Speaking time	8.2 sec.	6.9 sec.
Waiting time	42 sec.	39 sec.

Table 2: Comparison of Conditions with and without Familiarization Scenario

	1	2	3	4
Unique words/subject	81	89	83	67
Unique words/condition	296	344	270	219
Extra-lexical items, No. words	66	80	61	0
Extra-lexical items, No. sentences	74	205	138	0
(percent sentences)	(31)	(81)	(87)	(0)
Vocabulary errors	0	0	0	3.8
Other errors	3.7	4.2	1.8	0.6
Task Time (min)	37	43	22	24

Table 3: Comparison of Condition 1 (SRI-TI), 2 (Task Specificity), 3 (Familiarization Scenario), and 4 (Finite Vocabulary).

narios took significantly less time to complete the main task (23.2 vs. 39.9 minutes, $p < .01$) and used significantly fewer words to complete the task (276 vs. 146, $p < .01$) than subjects in the other two conditions. The difference between the number of utterances produced by the two groups was not significant, however (24.4 vs. 17.2, $p > .05$), while the number of words per utterance used by subjects in the training conditions was fewer (8.7 vs. 12.2, $p < .01$). Subjects in the familiarization conditions also received fewer error messages per utterance produced (.07 vs. .13) and asked fewer questions concerning the meanings of table headings (13% of all queries, compared to 25% for subjects with no familiarization scenario).

Condition 4: Finite Vocabulary

Earlier work concerning the vocabulary used by subjects and the percent of new words introduced in each session suggested that expert human-machine users could potentially adapt to a restricted vocabulary and still maintain efficiency [1]. In order to test whether subjects would adapt to a restricted vocabulary, we slightly modified our system to accept only a limited vocabulary from the subjects. The wizard's assistant, instead of being provided with a normal spell-checker, used a spell-checker that contained only a subset of approximately 1000 most frequently used words, based on the data released by TI in distributions 1-4 (prepilot data plus NIST Release 1). Subjects were made aware of this restriction in the instructions and were provided with a list of acceptable words. If they used a word outside the vocabulary, they were sent the message: "You have used a word outside the system's vocabulary. Try rephrasing your request." In all other respects, this "Fixed Vocabulary" Condition was identical to the Familiarization Condition (i.e., subjects in this condition were given a familiarization scenario and performed a constrained task).

If we compare the subjects who received a familiarization scenario but were unlimited in vocabulary and those who received a familiarization scenario but were limited to a 1000-word vocabulary, we find that the error messages received by the latter group for using out-of-vocabulary items is higher. During the familiarization session, they received an average of 2.0 error messages of this kind, and an average of 3.8 messages of this kind for the main task. When added to the other error messages they received, this gave them a slightly higher number of total error messages received than subjects in the comparable but unlimited-vocabulary condition (4.4 vs. 1.8). The mean number of error messages received by the group was not, however, different from the mean number of error messages received by subjects in either of the non-familiarization scenario conditions. In addition, there is evidence for the adaptation of subjects to a fixed vocabulary as indicated in Table 3. This table indicates that with a short familiarization session and consistent feedback one can dramatically affect the number of unique words used by the subject, the number outside a fixed set, and the number of sentences with such "extra-lexical" items, without increasing the total time to complete the task. The discrepancies between the number of "extra-lexical" items and the number of

sentences in which they occur arise because some subjects will use a given lexical item in many subsequent sentences once it has "worked".

Discussion

In addition to replicating the results released by TI, using a setup similar to TI's, we have shown the effect of altering various aspects of the experimental setup, including scenario specificity, subject familiarization and restricting the vocabulary.

We believe that our results indicate that we have succeeded in implementing a functional equivalent of the TI data collection system. The one major exception to this claim is the observed discrepancy in the word frequency distributions. This discrepancy can be remedied by avoiding any sample sentences from the domain while instructing subjects.

In assessing scenario specificity, we found no differences on either yield measures (time to complete task, utterances per task, words per task, etc.) or on quality measures (error message rates, cancellation rates) between subjects in the unconstrained task condition and those in the constrained (specific) task condition. In light of this, one might argue for adopting specific scenarios on the basis of the benefits gained by knowing subjects are interpreting the task the same way (in effect, are performing the same task) and by obtaining data useful for both analysis of isolated queries and of dialogue.

Our most significant results pertain to subject familiarization. In two separate experiments using two very different interfaces and procedures, we demonstrated the impact of subject familiarization with the system: subjects less familiar with the system produced longer utterances, needed more time to complete the task, and produced fewer utterances per subject hour. The time to familiarize subjects with the system (5 to 6 minutes) was short relative to the gains in subject efficiency (17 minutes saved on average in subject time to complete task).

Our Fixed Vocabulary Condition showed that subjects can adapt quickly to a restricted vocabulary without increasing task time: subjects in the Fixed Vocabulary Condition did not take longer to complete the task or to plan each utterance than those in the unlimited-vocabulary conditions, so the constraint doesn't appear to slow them down unnaturally or lower the yield of the experimental session. It is worth noting that these subjects showed significant improvement in the number of out-of-vocabulary error messages received during the main task (3.8 in 24.29 minutes) as compared to the training scenario (2.0 errors in 7.27 minutes). This supports the position that subjects can adapt to using a limited vocabulary. This result may be very important in the development of scalable technologies that will fit on a variety of platforms.

We found no systematic differences in the answers subjects provided to the questionnaire we presented to them after the session. The subjective experience of the subjects in the various conditions, then, seems to have been about the same.

The goals of designing an appropriate spoken language system can sometimes conflict with the goal of collecting data for evaluation of spoken database queries. That is, some major causes of errors (e.g., out-of-vocabulary items, out-of-domain queries) may disappear with a small amount of either detailed instruction or subject familiarization. However, we are convinced that it is possible to find ways of coordinating the two endeavors. For example, the needs of both dialogue analysis and of query-answer pairs for evaluation can be met using a more specific scenario; the needs of restricted vocabulary can be met by providing consistent feedback; and the large effect of subject familiarization can be addressed by spending a short time in the room with the subject to answer questions as the subject works on a task.

We plan to continue these experiments to help us design an appropriate human-machine interface. In our next set of experiments we will include a revised grammar for NLParse that reduces the number of words the wizard needs to produce by about 35% (on "cheapest" constructions it can reduce the number of words to about a quarter of the number that would be needed without the modification). Other experiments we are planning include the reformatting of tables sent by Oracle (the high percentage of queries concerning the meanings of various column headings indicate that much could be done to improve the user interface in this area), and some variations on the use of push-to-talk mechanism. We will also be running repeat subjects to test the effect of longer use of the system on the resulting data.

Acknowledgements

The authors gratefully acknowledge TI and Charles Hemphill, in particular, for helping us to get a license to NLParse, for providing much of the related software and data, and for helpful responses to our endless pleas for assistance. We also thank CMU for providing recording and playback software for the NEXT machine. This research was funded by DARPA under Office of Naval Research contract N00014-90-C-0085.

References

[1] J. Kowtko, P. Price and S. Tepper, "Data Collection and Analysis in the Air Travel Planning Domain," DARPA Speech and Natural Language Workshop, October 1989.

[2] C. Hemphill, J. Godfrey,, and G. Doddington, "The ATIS Spoken Language Systems Pilot Corpus," this volume.

Spontaneous Speech Effects in Large Vocabulary Speech Recognition Applications

John Butzberger, Hy Murveit, Elizabeth Shriberg, Patti Price

SRI International
Speech Research and Technology Program
Menlo Park, CA 94025

ABSTRACT

We describe three analyses on the effects of spontaneous speech on the recognition of continuous speech. We have found that: (1) spontaneous-speech effects significantly degrade recognition performance, (2) *fluent* spontaneous speech yields word accuracies equivalent to read speech, and (3) using spontaneous-speech training data can significantly improve performance for recognizing spontaneous speech. We conclude that word accuracy can be improved by explicitly modeling spontaneous effects in the recognizer and by using as much spontaneous speech training data as possible. Inclusion of read-speech training data, even within the task domain, does not significantly improve performance.

1. INTRODUCTION

Recognition of spontaneous speech is an important feature of database-query spoken-language systems (SLS). However, most speech recognition research has focused on acoustic and language modeling developed for recognition of read speech [1]. Read speech has been used extensively in the past for both training and testing speech recognition systems because it is significantly less expensive to collect than spontaneous speech, and because the lexical and syntactic content of the data can be controlled.

The multisite data collection effort [3] has provided a challenging corpus for research and development in the Airline Travel Information System (ATIS) domain. We have observed a significant increase in word error rate compared to the previous task domain, the read-speech naval Resource Management (RM) task [2,6]. Word error rates for RM systems have typically been in the 5% range, whereas ATIS word error rates have exceeded 10% [4], for comparable perplexities.

The speaking style typically exhibited in the RM domain had a very consistent rate and articulation, within and across sentences, and across speakers. There were no disfluencies, such as word fragments, hesitations, or self-edits, since utterances containing these effects were removed from the corpus. The utterances tended to be short and direct (3.3 seconds long, on average). No pause fillers (uh, um), false starts, repairs, or excessively long pauses occurred. The speakers were able to concentrate on speech production, rather than query formation or problem solving. Furthermore, the training and testing texts were generated using a fixed vocabulary, and with the same, known language model, which quite adequately represented the source and target languages.

The speaking style typically exhibited in the ATIS domain differs from that in the RM domain in all of the above aspects. The speaking rate is highly inconsistent, within utterances, across utterances within a session, and across sessions and speakers. The articulation is highly variable, with stressed forms of function words and reduced forms of content words typically not observed in read speech. The sentence lengths vary widely and are typically longer than RM sentences (7.5 seconds long, on average). Some words in ATIS sentences may not exist in the recognizer's lexicon, and an appropriate language model must be developed.

Most important, however, ATIS speech contains spontaneous effects and disfluencies: filled pauses, stressed or lengthened function words, false-starts and self-edits, word fragments, breaths, long pauses, and extraneous noises such as paper rustling and beeps. Data collected using systems containing automatic speech recognition and natural language components contain frequent occurrences of hyperarticulated words, elicited by the subjects in an attempt to overcome recognition or understanding errors [5]. Additionally, the data have been collected in normal office conditions (rather than in a soundproof booth), and recording quality and conditions vary across sites [3].

2. ERROR ANALYSIS

We begin by analyzing the errors that occurred in the February 1991 evaluation set of 148 Class-A sentences, for which our word error rate exceeded 18%. These sentences are examined because they are believed to be a particularly difficult sample of ATIS speech.

Phonetic alignments were automatically generated corresponding to both the reference and recognized word strings, and each utterance was listened to very carefully. We compared the acoustic and language model scores, and made a subjective judgment as to the likely source of the error (the acoustic model, the language model, the articulation quality of the segment, or other effects such as breaths, out-of-vocabulary words, or extraneous noise).

We found that 30% of the errors (Table 1) could be attributed to poor articulation or poorly modeled articulation (usually reductions, emphatic stress, or speaking-rate variations); 20% were due to out-of-vocabulary words or poor bigram probabilities; 20% were due to unmodeled pause-fillers (uh, um, breaths). The remaining 30% could not be attributed to any of the above, but were probably due to inadequate acoustic-phonetic modeling.

We see that 70% of the errors are due to effects observed in the ATIS domain, but not in the RM domain. If these errors were removed, we would approach an error rate typically seen in a comparable RM system (with a perplexity 60 word-pair grammar).

Corpus	Cause for Error	Portion
ATIS only	Poor Articulation	30%
	Vocabulary and Grammar	20%
	Pause Fillers	20%
ATIS and RM	Other	30%

Table 1: Summary of error sources for the Class-A Feb91 ATIS evaluation set (148 sentences).

3. READ VS. SPONTANEOUS SPEECH

To determine the impact of spontaneous versus read speaking styles on recognition performance given a fixed training condition, we constructed a recognition experiment with two test sets. The first set contained spontaneous speech utterances; the second set contained read versions of those same utterances, given later by the same subjects.

The training data consisted of RM, TIMIT, and pilot-corpus ATIS utterances (with the read-spontaneous and spontaneous test data held out). This left rather little ATIS-specific data for training, almost none of it spontaneous. The recognition was run without a grammar (perplexity 1025) to remove any corrective effects of the grammar, so that only the acoustic effect of the spontaneous speech could be evaluated. The spontaneous test sentences were categorized as either fluent or disfluent based on the existence of special markings in their corresponding SRO* files.

We found that the primary difference in error rates between the read and spontaneous test sets was due directly to disfluencies in the spontaneous speech (Table 2). *Nondisfluent spontaneous speech had the same error rate as read speech.* The disfluencies include pause-fillers, word fragments, overly lengthened or overly stressed function words, self-edits, mispronunciations, and overly long pauses. This list of disfluency types is derived from the special markings used in the SRO transcriptions. The observation that the nondisfluent spontaneous-speech error rate approaches the read-speech error rate is consistent with the fact that the test speech much more closely resembles the training data. The utterances in the training data were fluently and consistently articulated, just as was the nondisfluent spontaneous test utterances.

Characteristic	Number of Sentences	Word Error
Read	241	33%
Spontaneous	241	43%
Spontaneous - Disfluent	97	56%
Spontaneous - Fluent	144	32%

Table 2: Error rate versus speaking style. Read speech and fluent spontaneous speech have roughly equivalent error rates.

The breakdown of error rate versus disfluency type (Table 3) shows that a significant portion of the errors were due to filled pauses, long pauses, lengthenings, and stress. Sentences with these disfluencies had twice the word error rate of fluent speech. The filled-pause errors happened because there were no models for breath/uh/um events in this particular recognizer's lexicon. The stress and lengthening errors happened (most likely) because of the lack of sufficient observations of these events in the training data, and because of the lack of explicit models for these effects. The long pauses usually caused insertions within the pause regions neighboring the phrase-initial and phrase-final words.

From these observations, we conclude that more training data containing these effects would improve the match between the acoustic models and the spontaneous test speech, and therefore would improve the recognition performance. Furthermore, these effects should be explicitly modeled in the recognizer's lexicon, once sufficient training data are obtained. However, this process depends on the reliability of the SRO labeling across sites, which tends to be subjective and inconsistent.

*The SRO transcription contains a detailed description of all the acoustic events occurring in an utterance.

Disfluency Type	Number of Sents	Disfluency Causes Error
Self-Edit	7	71%
Filled Pause	24	92%
Long Pause	17	94%
Lengthening	36	81%
Stress	22	59%
Mispronunciation	2	100%
Fragment	5	100%

Table 3: Number of sentences afflicted with each disfluency type, and the percentage of occurrences where the disfluency causes an error.

4. TRAINING DATA VARIATIONS

Further evidence for the importance of modeling spontaneous-speech phenomena is found by manipulating the content of the training data sets that are used for acoustic-phonetic modeling. In this experiment, we compare the performance of spontaneous-speech recognition for different combinations of read, spontaneous, ATIS, and non-ATIS training subsets.

The training subsets (Table 4) consist of the standard RM and TIMIT training data, and read and spontaneous subdivisions of all the ATIS and MADCOW data available as of October 1, 1991. The "Breaths" corpus refers to an internally collected database of inhalations and exhalations, used to train a breath model, which is allowed to occur optionally between words during recognition. Much of the ATIS-read data were also collected internally at SRI.

Corpus	Size
ATIS-Read	7,932
ATIS-Spontaneous	6,745
TIMIT	4,200
Resource Management	3,990
Breaths	800

Table 4: Training data subsets, which are combined in various ways to determine the impact of read and spontaneous training data on recognition of spontaneous speech.

Recognition was conducted using a development test-set of 447 spontaneous MADCOW utterances [3], with a perplexity 20 bigram grammar trained on all the available spontaneous speech transcriptions (roughly 10,000 sentences). All of the experiments outlined below use discrete-distribution hidden Markov models (HMMs), and every training set combination includes the 800 breath utterances.

Using all the available ATIS and MADCOW data yielded a system with a word error rate of 9.6% (Table 5). Using only spontaneous ATIS speech reduced performance by only 6%, to 10.2% word error. Training with a roughly equivalent quantity of read ATIS speech increased the error rate significantly, by 58% to 15.2%. This suggests that having training data that are consistent in speaking mode with the test data can significantly improve performance. However, the effect of lexical and phonetic coverage in the training sets might be a factor in causing this performance difference. This issue is discussed in Section 5.

Training Set	Size	Error
ATIS-Read	8,732	15.2%
ATIS-Spontaneous	7,545	10.2%
ATIS-All	15,477	9.6%

Table 5: Training set variations for ATIS-only systems, showing how speaking-mode-consistent data improves performance.

We also look at the impact of using non-ATIS read speech for additional training data (Table 6). Using successively more training data gives the expected result, an improvement in performance. However, when using all the available data (RM, TIMIT, ATIS and MADCOW), the performance matches that of the system trained exclusively on ATIS and MADCOW data. Furthermore, the performance of the system trained using all the available read speech (16,922 sentences) performed much worse than the system trained only on spontaneous speech (7,545 sentences).

Training Set	Size	Error
TIMIT	5,000	26.9%
TIMIT + RM	8,990	20.5%
TIMIT + RM + ATIS-Read	16,922	14.6%
TIMIT + RM + ATIS-All	23,667	9.6%

Table 6: Training set variations using non-ATIS data, showing a drop in the error rate when ATIS-read data are added, and a further drop when ATIS-spontaneous data are added.

We can conclude from these experiments that having speaking-mode-consistent training data is more important than simply having a large quantity of training data. However, we cannot be certain that the phonetic content of the ATIS-spontaneous training set matches the development set better than the ATIS-read training set. That issue is addressed in Section 5.

We compared the errors of two different recognizers used on the same test set of spontaneous speech. Both recognizers were trained on a comparable number of utterances, but one recognizer was trained on read speech only (TIMIT+RM+ATIS-Read), and the other on spontaneous speech only (ATIS-Spontaneous). We found that substitutions of one function word for another form a significant portion of the errors in both test sets, and in roughly the same proportions. However, there were significantly fewer substitutions of content words for other content words for the recognizer trained on spontaneous speech than for the recognizer trained on read speech.

Similarly, the recognizer trained on spontaneous speech manifested significantly fewer errors in substitution of a pause filler for a function word. Homophone errors, which can lead to understanding errors, formed a significant portion of the errors in the recognizer trained on read speech, although almost none of these appeared for the recognizer trained on spontaneous speech. We believe that this is because many words that can be homophonous in read speech ("for" "four" and "to" "two", for example) are no longer homophones in spontaneous speech ("fer" "four" and "tuh" "two").

5. Phonetic Coverage Analysis

One potential reason for the dramatic performance variations could be that the phonetic content of the development test set is better covered by the ATIS-Spontaneous subset than by the ATIS-Read subset. In this section, we attempt to disprove that theory, giving further strength to the argument that speaking-mode consistency is the primary factor affecting performance.

We reason that the more detailed (more context-dependent) acoustic-phonetic models there are available for testing, the more adequate the training data have been in representing this dimension (the better the phonetic coverage). Therefore, for this analysis, we determine the average context level (or detail) of HMM states that each frame of test data visits during recognition. The average is computed by assigning an integer-valued number to each model type (increasing as context level increases), then computing the percentage of all frames of data visiting states corresponding to a particular level of context.

The series of context-dependent model types used in the DECIPHER system is listed in Table 7. A model with a particular context level will be generated by the DECIPHER trainer if there are sufficient data to train that model.

Model Type	Context Level
Monophone	1
Left-general biphone	2
Right-general biphone	2
Left biphone	3
Right biphone	3
General triphone	4
Left-general triphone	5
Right-general triphone	5
Triphone	6
Word-specific	7

Table 7: Assignments of an integer-valued context level to each context-dependent model type. Models with increasing detail are assigned higher context level values.

The expectation is that the higher the average context level encountered during recognition, the better the performance. This trend is indeed captured in Table 8, where the system with the least task-specific training data (TIMIT) had the least average context level (and the lowest performance), and the system with the most training data (TIMIT+RM+ATIS-All) had the highest average context level (and the highest performance).

The important point is that the average context level of the best-trained read speech system (TIMIT+RM+ATIS-Read) was roughly equal to that of the best spontaneous-only system (ATIS-Spontaneous), but the error rate was significantly higher (14.6% versus 10.2%, respectively). This suggests that although models of equivalent detail are being used for recognition, the performance difference is due to the spontaneous speaking mode of the training set, which is consistent with the speaking mode of the test set.

Training Sets	Error Rate	Context Level
TIMIT+RM+ATIS-All	9.6%	6.31
ATIS-All	9.6%	6.26
ATIS-Spontaneous	10.2%	6.03
TIMIT+RM+ATIS-Read	14.6%	6.14
ATIS-Read	15.2%	5.96
TIMIT+RM	20.5%	5.06
TIMIT	26.9%	4.56

Table 8: Context level versus word error, indicating that despite similar model detail (context level), the spontaneous-speech-trained system significantly outperforms the best read-speech-trained system.

6. CONCLUSION

These studies have convinced us of the importance of using as much spontaneous speech material as possible in training our system. Furthermore, we have found that spontaneous speech effects can significantly degrade recognition performance, although *fluent* spontaneous speech yields word accuracies equivalent to those of read speech.

Word accuracy can be improved by using as much spontaneous-speech training data as possible, and by explicitly modeling such effects in the recognizer's lexicon (such as optional interword breath and pause-filler models). Inclusion of read-speech training data did not significantly improve performance, given that the phonetic coverage of the training sets were roughly equivalent.

Acknowledgments

We gratefully acknowledge support for this work from DARPA through the Office of Naval Research Contract N00014-90-C-0085. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the government funding agencies.

REFERENCES

1. P. Price., W. Fisher, J. Bernstein, and D. Pallet, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *Proc. ICASSP*, 1988.

2. D. Pallet, J. Fiscus, and J. Garofolo, "DARPA Resource Management Benchmark Test Results June 1990," *Proc. DARPA Speech and Natural Language Workshop*, R. Stern (ed.), Morgan Kaufmann, 1990.

3. MADCOW, "Multi-Site Data Collection for a Spoken Language System," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

4. H. Murveit, J. Butzberger, and M. Weintraub, "Performance of SRI's DECIPHER Speech Recognition Systems on DARPA's ATIS Task," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

5. E. Shriberg, E. Wade, and P. Price, "Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

6. H. Murveit, J. Butzberger, and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS Systems," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

THE DECIPHER SPEECH RECOGNITION SYSTEM

Michael Cohen, Hy Murveit, Jared Bernstein,
Patti Price and Mitch Weintraub

SRI International
Menlo Park, CA 94025

Abstract

DECIPHER is SRI's HMM-based speaker-independent continuous speech recognition system. DECIPHER performs well on the speaker-independent DARPA resource management task, as described in last year's ICASSP Proceedings [10]. To determine whether speaker-specific acoustic and phonological adaptation can further improve performance, the current paper describes DECIPHER's performance on a speaker-dependent task.

1. Introduction

The Speech Research Program at SRI International has designed and implemented several speech recognition systems in the last six years. SRI's current large-vocabulary, continuous-speech system, DECIPHER, is based on a hidden Markov model (HMM) approach and was designed to achieve high word accuracy in a speaker-independent mode. It has been trained and tested on DARPA's Resource Management database [9]. The DECIPHER system was described at last year's ICASSP meeting [10]. That paper presented results showing that speaker-independent recognition performance could be improved by incorporating certain kinds of linguistic knowledge into the Markov model framework, including cross-word coarticulatory modeling and detailed modeling of phonological variation.

This paper presents the results of a series of experiments that tested acoustic and phonological adaptation of the DECIPHER system to the pronunciations of a single speaker in a speaker-*dependent* task.

2. The DECIPHER System

The DECIPHER system uses an HMM framework similar to that used in a number of other systems [2, 7, 8]. The overall structure of such a system is well described in [7]. The overall structure of SRI's DECIPHER system is shown in Figure 1.

DECIPHER's front end samples an analog acoustic signal 16,000 times per second after passing the signal through a 6.4 Khz low-pass, anti-aliasing filter with 0.95 pre-emphasis. Signal analysis starts with a 512-point discrete Fourier transform (DFT) calculated every 10 msec on a 25.6 msec Hamming window. Four discrete acoustic features are calculated every 10 msec. The features are based on a 13-dimensional cepstral transform of the logarithms of the energies in 25 overlapping filters (approximately equally spaced on the mel scale) in the range from 100 Hz to 6400 Hz. An optional noise-robust spectral estimation process is described in [6] in this volume.

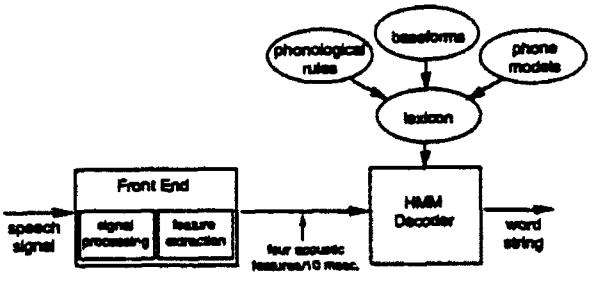


FIGURE 1: The DECIPHER System

The phonetic models in the DECIPHER system are discrete density 3-state hidden-Markov models. There are four discrete densities per state, one for each of the four acoustic features produced by the front end. Word models are directed graphs of phonetic models (combining context-independent and context-dependent phonetic models). The lexical graph for a vocabulary item is generated by the application of a set of phonological rules to a baseform pronunciation (similar to previous efforts at modeling multiple pronunciations [4]). The modeling of multiple pronunciations in the DECIPHER system differs from previous efforts in two important respects:

[1] A new technique for developing phonological rule sets was used, with the goal of maximizing the coverage of the pronunciations found in a corpus of speech while minimizing the size of networks.

[2] A new algorithm was used to estimate the probabilities of alternate pronunciations. The new algorithm defines sub-word units which can share training data based on equivalence classes of nodes.

These two techniques are described in the following two sections.

3. Developing Phonological Rule Sets

Previous efforts to model multiple pronunciations of words have suffered because many new parameters were introduced which had to be estimated with a fixed amount of training data. The approach to rule set development SRI uses has the goals of maximizing the coverage of observed forms in a corpus of speech while minimizing the size of the networks, and therefore minimizing the number of parameters which need to be estimated.

A number of software tools were developed which allow the measurement of the coverage of pronunciations in a corpus as well as overgeneration (generation of pronunciations not used), both for a full rule set and for the individual rules in a rule set. These tools can be used to optimize the definition of the contextual constraints of individual rules, as well as the choice of rules to include in a rule set.

The development of phonological rule sets proceeds as follows:

[0] Start with a lexicon of base forms, a corpus of pronunciations, and (optionally) a phonological rule set (i.e., we can start with an existing rule set and refine it, or start with just baseforms).

[1] Measure coverage of output forms (resulting from the application of current rules, if any, to baseforms) on observed pronunciations. Get diagnostic information on uncovered pronunciations.

[2] Write rules to cover pronunciations.

[3] Measure coverage and overgeneration of individual rules. Analyze and refine contextual specifications of rules based on individual rule diagnostics.

[4] Repeat from step 1 to achieve high coverage rule set.

Using the method outlined above, we have been able to develop a phonological rule set with significantly higher coverage and significantly lower overgeneration than rule sets developed by more traditional methods both at SRI and elsewhere [3].

4. Estimating the Probabilities of Alternative Pronunciations

Previous efforts to model multiple pronunciations of words have suffered because the unlikely pronunciations (not previously modeled) caused false alarms. This was a problem because the systems lacked accurate estimates of the probabilities of the many pronunciations modeled. Achieving accurate estimates is difficult because current databases for training recognition systems have too few occurrences of all but the most frequent words to make accurate estimates.

In order to reliably estimate pronunciation probabilities for words which don't happen frequently enough to provide adequate training data, it is necessary to tie together sub-word units which do happen frequently. Thus, reliable probabilities can be estimated for these sub-word units, which can then be concatenated to form estimates for word pronunciations. Because extended context can play an important role in determining the allophonic form of a segment in a word, we want to tie together the largest units possible that have adequate training data, in order to capture the greatest amount of contextual information. We have developed an approach which attempts to automatically determine the best grouping of sub-word units into node-equivalence classes for common training.

In the DECIPHER system, the training of pronunciation probabilities is incorporated into the training of the HMM models using the forward-backward algorithm. The forward-backward algorithm provides estimates of the number of transitions for each arc at the end of each iteration through the training data. The estimated transitions for arcs which correspond to arcs in pronunciation networks are used to reestimate pronunciation probabilities allowing arcs to share training samples when they occur in the same node-equivalence class, as defined above.

We have shown improvements in speaker-independent performance using the rule set development and node-equivalence class training techniques outlined above [10]. The next section reports the evaluation of these techniques on a speaker-dependent database.

5. Speaker-Dependent Phonology

A set of experiments were performed in which pronunciation models were adapted to individual speakers. Initially, each speaker started with a set of pronunciation networks which resulted from the application of a phonological rule set, developed using the method described above, to a set of baseforms. The mean number of pronunciations represented per word with these networks was approximately 35. These networks were then trained separately for each speaker in the speaker-dependent test set. The training set for each speaker included 600 read sentences (the DARPA speaker-dependent resource management training set). Two iterations of the forward-backward algorithm were run, and the node-equivalence class algorithm referred to above was used to estimate speaker specific pronunciation probabilities for these networks. The networks were then pruned by removing low probability arcs, using an algorithm that includes constraints to prevent the creation of disconnected components of word networks and to avoid the creation of word models which can't connect to other words due to cross-word phonological constraints. In addition, node types with less than a specified minimum

number of training instances were constrained so that only the most likely arc was left after pruning.

These pruned speaker-dependent word networks had an average of approximately four pronunciations per word. An additional two iterations of the forward-backward algorithm were then run in order to train the acoustic HMM models with the pruned speaker-dependent word networks.

Tests were run with the DARPA 1000-word resource-management database using both the DARPA February 89 speaker-dependent test set and the 100-speaker development set. The DARPA perplexity-60 word pair grammar was used. Results are shown in Table I. The single networks were derived by pruning out all but the single, most likely, path in all of the word networks after training pronunciation probabilities using the node-equivalence class training algorithm. The multiple pronunciation networks were pruned, as described above, until there were an average of approximately four pronunciations per word. Table I compares performance using networks with pronunciation probabilities based on a speaker-independent training set and a speaker-dependent training set. (Only the training and pruning of pronunciation networks was varied for these runs - in all cases the acoustic HMM models were trained speaker-dependently.) Percent word correct was measured as

$$1 - \frac{insertions + deletions + substitutions}{total}$$

where total = number of words in the correct sentences

phonological training	networks	dev set	Feb 89 set
SI	single	97.5	97.0
SI	multiple	97.6	97.4
SD	single	97.6	97.4
SD	multiple	97.8	97.7

Table I: Speaker-dependent phonology.

As can be seen in Table I, a reduction in error rate of 12% was achieved for one test set and 23% for another test set going from speaker-independently determined single most-likely pronunciations to speaker-dependently determined multiple pronunciations. It can be seen that part of that gain can be achieved with speaker-specific adaptation of pronunciation networks, and part with the representation of multiple pronunciations. In all four cases shown, going from single pronunciations to multiple pronunciations improved performance, and going from speaker-independent to speaker-dependent phonological training improved performance.

6. Discussion

The results shown here suggest that:

[1] Speaker specific phonological training can improve recognition performance, both for single and multiple pronunciation systems.

[2] Multiple pronunciation models can improve the performance of a speaker-dependent system.

In both cases, the improvements observed were small, but consistent. A larger speaker-specific training set would be likely to improve the results reported here. With a larger training set, bushier word networks could be used while maintaining the accuracy of the estimates of pronunciation probabilities, as well as the estimates of the acoustic parameters of the HMM models.

All the results presented in this paper are based on experiments that both trained and tested the DECIPHER system on carefully collected, read speech. In the future, we intend to evaluate these techniques on goal-directed, spontaneous speech. These techniques are likely to become more important when DECIPHER is used with spontaneous speech where there is significantly increased in phonological reduction and deletion. [1,5].

References

[1] Bernstein, J., Baldwin, G., Cohen, M., Murveit, H. and Weintraub, M., "Phonological Studies for Speech Recognition," *Proceedings: DARPA Speech Recognition Workshop*, pp. 41-48, February, 1986.

[2] Chow, Y.L., Dunham, M.O., Kimball, O., Krasner, M., Kubala, F., Makhoul, J., Roucos, S., Schwartz, R.M., "BYBLOS: The BBN Continuous Speech Recognition System," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 89-92, April, 1987.

[3] Cohen, M.H., *Phonological Structures for Speech Recognition*, PhD thesis, Computer Science Department, UC Berkeley, April 1989.

[4] Cohen, P.S. and Mercer, P.L., "The Phonological Component of an Automatic Speech Recognition System," in *Speech Recognition*, R. Reddy, ed., Academic Press, New York, p. 275-320.

[5] Dalby, J., *Phonetic Structure of Fast Speech in American English* PhD thesis, Linguistics Department, Indiana University, December, 1984.

[6] Erell, A., and Weintraub, M., "Estimation Using Log-Spectral Distance Criterion for Noise Robust Speech Recognition," this volume.

[7] Lee, K.F., *Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*, PhD thesis, Computer Science Department, Carnegie Mellon University, April 1988.

[8] Paul, D., "Site Report and Benchmark Tests," presented at *DARPA Speech Recognition Workshop*, June, 1988.

[9] Price, P., Fisher, W.M., Bernstein, J. and Pallet, D.S., "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 651-654, April, 1988.

[10] Weintraub, M., Murveit, H., Cohen, M., Price, P., Bernstein, J., Baldwin, G., and Bell, D., "Linguistic Constraints in Hidden Markov Model Based Speech Recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May, 1989.

CSR Corpus Development

George R. Doddington

SRI International
Menlo Park, CA

ABSTRACT

The CSR (Connected Speech Recognition) corpus represents a new DARPA speech recognition technology development initiative to advance the state of the art in CSR. This corpus essentially supersedes the now old Resource Management (RM) corpus that has fueled DARPA speech recognition technology development for the past 5 years. The new CSR corpus supports research on major new problems including unlimited vocabulary, natural grammar, and spontaneous speech. This paper presents an overview of the CSR corpus, reviews the definition and development of the "CSR pilot corpus", and examines the dynamic challenge of extending the CSR corpus to meet future needs.

OVERVIEW

Common speech corpus development and evaluation received major emphasis from the very beginning of the DARPA speech recognition program. At that time, a set of common corpora were defined to serve the needs of the research community. This resulted in the development of the TIMIT speech corpus, which was collected from a large number of subjects and intended to support basic research in acoustic-phonetic recognition technology. The Resource Management (RM) corpus, collected from fewer subjects but representing an application of interest to DARPA, provided the greatest focus of interest in technology throughout the research community. In the course of R&D using these two corpora, the first serious research and advances toward speaker-independent speech recognition were achieved.

Although the RM corpus served its intended purpose well, technology advances came to make its limitations painfully obvious. The language was artificial and limited, the speech was read and therefore unnatural, and the corpus completely avoided the central issue of understanding the meaning of the spoken utterances. In response to these limitations and to rapid advances in the performance of speech recognition technology on this RM task, a new research initiative was formed by combining speech recognition and natural language understanding tasks in a spoken language system (SLS) program.

The SLS program took shape with the definition of the Airline Travel Information System (ATIS) task, a database query task which supports research in both speech recognition and natural language. The ATIS corpus (corpora) is currently being collected to provide the experimental data for developing SLS technology. This ATIS corpus exhibits several desirable features regarding the speech recognition problem that were found lacking in the RM corpus. These features are namely the use of spontaneous goal-directed speech and the consequent use of a natural grammar and an open unrestricted vocabulary.

Although the ATIS corpus provides the kind of speech data desired by the speech recognition research community and required to address important problems in the application of speech recognition to real tasks, there is one unfortunate shortcoming of this corpus. This is that the cost and effort of collecting the data is too great to support the massive data requirements for advances in speech recognition technology. Some way of improving the efficiency and productivity of data collection was needed in order to support further advances in speech recognition technology. This need was the primary motivation for the creation of the CSR research initiative and its related CSR corpus.

The CSR research initiative, along with the CSR corpus development effort, was created in order to provide better support for advances in the state of the art in large vocabulary CSR. The primary focus in the CSR initiative has been on the design and development of a CSR speech corpus which is required to fuel the research and through which the research might be productively directed. Primary objectives of the CSR corpus have been to increase the realism of the speech data and at the same time to maximize the efficiency of collecting that data. Efficiency has been viewed as of paramount importance because it is generally believed that significant advances in speech recognition technology will require more comprehensive models of speech and correspondingly more massive quantities of speech data with which to train them.

Janet Baker was the principal champion and designer of the CSR corpus, working as the chair of a CSR corpus design committee. This committee dealt with a large and diverse set of research interests and corpus needs, which made the

task of designing a satisfactory corpus extremely difficult. For example, the desire to collect spontaneous speech was in direct opposition to the need to make corpus development efficient (because spontaneous speech requires a generally painstaking and expensive transcription task, whereas read speech can be transcribed far more efficiently and even largely automatically). [1]

Major Corpus Design Decisions

- Read speech versus spontaneous speech: On the issue of spontaneous speech, it was decided that the majority of the corpus (and in particular the majority of the training data) should be read speech, for economic reasons, whereas the majority of the test data (which comprises a small fraction of the total data) should be spontaneous speech. The reason for these decisions is that it was felt that large amounts of read speech would provide greater training benefits than smaller amounts of spontaneous speech, while using spontaneous speech for testing would better validate the technology for a relatively small increase in cost.

- Prompting text: Probably the most significant decision regarding the CSR corpus was the decision to work initially with the Wall Street Journal (WSJ). This decision was influenced by the richness of the WSJ language and by the existence of a preexisting and very large (50 million word) corpus of WSJ text (as part of the ACL-DCI effort). All of the read speech data is currently being collected using prompts derived from the WSJ. The spontaneous speech data is being collected using a news reporting dictation paradigm that simulates the WSJ dictation scenario.[2]

- Verbalized punctuation: In dictation, which is the nominal target application for the CSR technology development effort, dictation users typically say punctuation such as "comma" and "period" so as to aid in the proper punctuation of the dictated document. Therefore, in order to improve the verisimilitude of the CSR corpus, a strong opinion was voiced that such verbalized punctuation (VP) be included in the prompting text. Opposed to this view was the opinion that such predetermined VP may not represent realistic VP, may limit research on automatic punctuation, may restrict the task and perplexity, may unduly burden the corpus with VP words, and may present a difficult and artificial reading task to users. As a result, a compromise position was taken in which half of the corpus was collected in VP mode and half in non-VP mode.

- Speaker-independence: The CSR corpus, although directed primarily toward speaker-independent recognition, also supports research into speaker dependent recognition. Approximately half of the pilot corpus is dedicated to speaker-dependent work

- Microphone independence: The primary microphone is the traditional Sennheiser model HMD-414. In addition, all data were collected also with a secondary microphone. Previously, this second microphone was a single far-field pick-up microphone, such as the desktop Crown model PZM-6FS. The CSR pilot corpus represents a departure from this practice and a first attempt at true microphone-independent recognition (in much the same spirit as speaker-independent recognition) by using one of many different microphones for the alternate (secondary) speech channel.

- Transcription: For the CSR pilot corpus, the original source text was preprocessed to produce a string of words that represented as well as practical the string of words that would result from reading the source text. This word string was then presented to the subject as the prompting text. This approach provided a very efficient transcription mechanism, because the prompting text could automatically be used as the transcription (except when the subject made errors in reading). Also, the language model, although perhaps a bit unnatural to the extent that the prompt string doesn't represent the statistics of the true language model, can be more easily and comprehensively estimated by preprocessing large volumes of text rather than by transcribing relatively small amounts of speech data.

The CSR Corpus Coordinating Committee

The charter of the CSR Corpus Coordinating Committee (CCCC) is to coordinate CSR corpus development and to resolve issues which arise in CSR corpus development and evaluation. There are currently 12 members of the CCCC, namely:

Janet Baker, Dragon
Jordan Cohen, IDA
George Doddington (chairman)

1. The design of the CSR pilot corpus is described in detail in the paper by D. Paul and J. Baker in this workshop's proceedings entitled "The Design for the Wall Street Journal-based CSR Corpus".
2. The spontaneous speech data collection effort is described in detail in the paper by J. Bernstein and D. Danielson in this workshop's proceedings entitled "Spontaneous Speech Collection for the CSR Corpus.

Francis Kubala, BBN
Dave Pallett, NIST
Doug Paul, Lincoln Labs
Mike Phillips, MIT
Michael Picheny, IBM
Raja Rajasekaran, TI
Xuedong Huang, CMU
Mitch Weintraub, SRI
Chin Lee, AT&T

This committee was formed at the SLS coordinating committee meeting in October 1991. Since that time the committee has met ten times, mostly via teleconference. CCCC activities have included:

- Definition of procedures for microphone gain adjustment and calibration.
- Definition of procedures for transcribing the speech data.
- Monitoring progress in speech data collection and transcription.
- Definition of the data distribution schedule and format.
- Definition of procedures for evaluation of vocabulary/speaker adaptive systems.
- Definition of procedures for scoring.
- Definition of recommended baseline performance evaluations.

The CSR pilot corpus

One of the primary motivations for creating the CSR task and corpus was to provide a sufficiently large corpus of data to properly support advances in speech recognition technology. This implies a very large effort, with many hundreds of hours of speech data being collected. Given the massive effort required, and appreciating the untried nature of many of the corpus parameters, it was decided that a pilot corpus should be collected first to determine the correctness of the many corpus design decisions and to allow modifications of these as necessary.

The CSR pilot corpus is described in a companion paper in these proceedings entitled "The Design for the Wall Street Journal-based CSR Corpus" by D. Paul and J. Baker. This corpus provides for the development and evaluation of both speaker-independent (SI) and speaker-dependent (SD) recognition. It uses the now-standard DARPA corpus approach of providing a three-part corpus: speech data for training the speech recognition system ("TRAINING"), speech data for developing and optimizing the recognition decision criteria ("DEVELOPMENT TEST"), and speech data for performing the formal performance evaluation ("EVALUATION TEST").

The CSR February 1992 dry run evaluation

The recommended baseline performance evaluations were defined by selection of training data set(s), testing data set(s), recognition conditions (vocabulary and language model), and scoring conditions. In the course of discussion on these issues it became clear that consensus was not possible on definition of a single set of evaluation conditions. This was in addition to the distinct differences between speaker-dependent (SD) and speaker-independent (SI) evaluation data and conditions. Some committee members felt that there should be no constraint on training material, to allow as much freedom as possible to improve performance through training data. Others believed strongly that calibration of performance improvement was paramount and therefore all sites should be required to use a single baseline set of training data. In the end, the committee was able only to identify a number of different training and test conditions as "recommended" alternatives for a baseline evaluation.

For training the recommended SI training corpus comprised 7240 utterances from 84 speakers. The recommended SD training corpus comprised the 600 training sentences for each of the 12 SD speakers. For the large-data speaker-dependent (LSD) training condition, the recommended SD training corpus comprised the 2400 training sentences for each of the 3 LSD speakers.

For testing there were a total of 1200 SI test utterances and 1120 SD test utterances. These data comprised, similarly and separately for SI and SD recognition, approximately 400 sentences constrained to a 5000-word vocabulary, 400 sentences unconstrained by vocabulary, 200 sentences of spontaneous dictation, and these 200 sentences as read later from a prompting text.

The vocabulary and language models used for the above-defined test sets were either unspecified (for the spontaneous and read versions of the spontaneous dictation), or were the 5000-word vocabulary and bigram grammar as supplied by Doug Paul from an analysis of the preprocessed WSJ corpus. (Actually, two different sets of bigram model probabilities were used, one modeling verbalized punctuation and one modeling nonverbalized punctuation. These two were used appropriately for the verbalized and nonverbalized punctuation portions of the test sets, respectively.)

Given the rather massive computational challenge of training and testing in such a new recognition domain, with larger vocabulary and greater amount of test data, not all of the test material was processed by all of the sites performing evaluation. Also, because of the variety of training and evaluation conditions, few results were produced that could be compared across sites. Two test sets, however, were evaluated on by more than a single site. Two sites produced results on the SD 5000-word VP test set (Dragon and Lincoln), and three sites produced results on the SI 5000-word

VP test set (CMU, Lincoln, and SRI). These results are given in a companion paper on "CSR Pilot Corpus Performance Evaluation" by David Pallett.

Future CSR corpus effort and issues

Several issues have been identified that bear on the CSR corpus and on potential changes in the design of the corpus:

- Verbalized punctuation. There is a significant argument to discontinue verbalized punctuation, for several reasons: It doubles the number of language models and test sets and thus the number of evaluation conditions. It is artificial in the sense that it is statistically unlike normal dictation, it is more difficult for many subjects to read, and it seems superfluous to the development of the underlying speech recognition technology.

- Preprocessed prompting text. There is argument to prompt the user with the natural unpreprocessed text from the WSJ rather than with the preprocessed word strings as produced by the text preprocessor. The reason is that the word strings do not represent the actual statistics of natural speech (see the companion paper by Phillips et. al entitled "Collection and Analyses of WSJ-CSR Data at MIT").

- Spontaneous speech. There is argument that the current paradigm for collecting spontaneous speech is not adequately refined to represent those aspects of spontaneous speech that are important in actual usage, and that spontaneous speech should remain in an experimental and developmental mode during the next CSR corpus phase.

- Adaptation. Speaker adaptation and adaptation to the acoustical environment has emerged as a major interest. It is clear that adaptive systems must be accommodated in the next phase of the CSR corpus.

- CSR corpus development effort. It is acknowledged that the CSR corpus development effort is a key activity in the support and direction of CSR research, and that this effort therefore requires program continuity and should not be treated as an occasional production demand that can be easily started and stopped.

These issues are currently under debate in the CCCC, and the next installment of the CSR corpus, to be called the CSR corpus, phase two, will no doubt reflect a continued distillation of opinion on these issues.

GEMINI: A NATURAL LANGUAGE SYSTEM FOR SPOKEN-LANGUAGE UNDERSTANDING*

John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
Internet: dowding@ai.sri.com

1. INTRODUCTION

Gemini is a natural language (NL) understanding system developed for spoken language applications. This paper describes the details of the system, and includes relevant measurements of size, efficiency, and performance of each of its components.

In designing any NL understanding system, there is a tension between robustness and correctness. Forgiving an error risks throwing away crucial information; furthermore, devices added to a system to enhance robustness can sometimes enrich the ways of finding an analysis, multiplying the number of analyses for a given input, and making it more difficult to find the correct analysis. In processing spoken language this tension is heightened because the task of speech recognition introduces a new source of error. The robust system will attempt to find a sensible interpretation, even in the presence of performance errors by the speaker, or recognition errors by the speech recognizer. On the other hand, a system should be able to detect that a recognized string is not a sentence of English, to help filter recognition errors by the speech recognizer. Furthermore, if parsing and recognition are interleaved, then the parser should enforce constraints on partial utterances.

The approach taken in Gemini is to constrain language recognition with fairly conventional grammar, but to augment that grammar with two orthogonal rule-based recognition modules, one for glueing together the fragments found during the conventional grammar parsing phase, and another for recognizing and eliminating disfluencies known as "repairs." At the same time, the multiple analyses arising before and after all this added robustness are managed in two ways: first, by highly constraining the additional rule-based modules by partitioning the rules into preference classes, and second, through the addition of a postprocessing parse preference component.

Processing starts in Gemini when syntactic, semantic, and lexical rules are applied by a bottom-up all-paths *constituent* parser to populate a chart with edges containing syntactic, semantic, and logical form information. Then, a second *utterance* parser is used to apply a second set of syntactic and semantic rules that are required to span the entire utterance. If no semantically acceptable utterance-spanning edges are found during this phase, a component to recognize and correct certain grammatical disfluencies is applied. When an acceptable interpretation is found, a set of parse preferences is used to choose a single best interpretation from the chart to be used for subsequent processing. Quantifier scoping rules are applied to this best interpretation to produce the final logical form, which is then used as input to a query-answering system. The following sections describe each of these components in detail, with the exception of the query-answering subsystem, which is not described in this paper.

In our component-by-component view of Gemini, we provide detailed statistics on each component's size, speed, coverage, and accuracy. These numbers detail our performance on the subdomain of air-travel planning that is currently being used by the ARPA spoken language understanding community (MADCOW, 1992). Gemini was trained on a 5875-utterance dataset from this domain, with another 688 utterances used as a blind test (not explicitly trained on, but run multiple times) to monitor our performance on a dataset on which we did not train. We also report here our results on another 756-utterance fair test set that we ran only once. Table 1 contains a summary of the coverage of the various components on both the training and fair test sets. More detailed

*This research was supported by the Advanced Research Projects Agency under Contract ONR N00014-90-C-0085 with the Office of Naval Research. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency of the U.S. Government.

explanations of these numbers are given in the relevant sections.

	Training	Test
Lexicon	99.1%	95.9%
Syntax	94.2%	90.9%
Semantics	87.4%	83.7%
Syntax (repair correction)	96.0%	93.1%
Semantics (repair correction)	89.1%	86.0%

Table 1: Domain Coverage by Component

2. SYSTEM DESCRIPTION

Gemini maintains a firm separation between the language- and domain-specific portions of the system, and the underlying infrastructure and execution strategies. The Gemini kernel consists of a set of compilers to interpret the high-level languages in which the lexicon and syntactic and semantic grammar rules are written, as well as the parser, semantic interpretation, quantifier scoping, repair correction mechanisms, and all other aspects of Gemini that are not specific to a language or domain. Although this paper describes the lexicon, grammar, and semantics of English, Gemini has also been used in a Japanese spoken language understanding system (Kameyama, 1992).

2.1. Grammar Formalism

Gemini includes a midsized constituent grammar of English (described in section 2.3), a small utterance grammar for assembling constituents into utterances (described in section 2.7), and a lexicon. All three are written in a variant of the unification formalism used in the Core Language Engine (Alshawi, 1992) .

The basic building block of the grammar formalism is a category with feature constraints. Here is an example:

```
np:[wh=ynq,case=(nom∨acc),
    pers_num=(3rd∧sg)]
```

This category can be instantiated by any noun phrase with the value ynq for its wh feature (which means it must be a wh-bearing noun phrase like *which book*, *who*, or *whose mother*), either acc (accusative) or nom (nominative) for its case feature, and the conjunctive value 3rd∧sg (third and singular) for its person-number feature. This formalism is related directly to the Core Language Engine, but more conceptually it is closely related to that of other unification-based grammar formalisms with a context-free skeleton, such as PATR-II (Shieber et al., 1983), Categorial Unification Grammar (Uszkoreit, 1986), Generalized Phrase-Structure Grammar (Gazdar et al., 1982), and Lexical Functional Grammar (Bresnan, 1982).

Gemini differs from other unification formalisms in the following ways. Since many of the most interesting issues regarding the formalism concern typing, we defer discussing motivation until section 2.5.

- Gemini uses typed unification. Each category has a set of features declared for it. Each feature has a declared value space of possible values (value spaces may be shared by different features). Feature structures in Gemini can be recursive, but only by having categories in their value space; so typing is also recursive. Typed feature structures are also used in HPSG (Pollard and Sag, in press). One important difference with the use in Gemini is that Gemini has no type inheritance.

- Some approaches do not assume a *syntactic skeleton* of category-introducing rules (for example, Functional Unification Grammar (Kay, 1979)). Some make such rules implicit (for example, the various categorial unification approaches, such as Unification Categorial Grammar (Zeevat, Klein, and Calder, 1987)).

- Even when a syntactic skeleton is assumed, some approaches do not distinguish the category of a constituent (for example, np, vp) from its other features (for example, pers_num, gapsin, gapsout). Thus, for example, in one version of GPSG, categories were simply feature bundles (attribute value structures) and there was a feature MAJ taking values like N,V,A, and P which determined the major category of constituent.

- Gemini does not allow rules schematizing over syntactic categories.

2.2. Lexicon

The Gemini lexicon uses the same category notation as the Gemini syntactic rules. Lexical categories are types as well, with sets of features defined for them. The lexical component of Gemini includes the lexicon of base forms, lexical templates, morphological rules, and the lexical type and feature default specifications.

The Gemini lexicon used for the air-travel planning domain contains 1,315 base entries. These expand by morphological rules to 2,019. In the 5875-utterance training set, 52 sentences contained unknown words (0.9%), compared to 31 sentences in the 756-utterance fair test set (4.1%).

2.3. Constituent Grammar

A simplified example of a syntactic rule is

```
syn(whq_ynq_slash_np,
  [ s:[sentence_type=whq, form=tnsd,
       gapsin=G, gapsout=G],
    np:[wh=ynq, pers_num=N],
    s:[sentence_type=ynq, form=tnsd,
       gapsin=np:[pers_num=N],
       gapsout=null]]).
```

This syntax rule (named `whq_ynq_slash_np`) says that a sentence (category `s`) can be built by finding a noun phrase (category `np`) followed by a sentence. It requires that the daughter `np` have the value `ynq` for its `wh` feature and that it have the value `N` (a variable) for its `person-number` feature. It requires that the daughter sentence have a category value for its `gapsin` feature, namely an `np` with a person number value `N`, which is the same as the person number value on the wh-bearing noun phrase. The interpretation of the entire rule is that a gapless sentence with `sentence_type whq` can be built by finding a wh-phrase followed by a sentence with a noun phrase gap in it that has the same person number as the wh-phrase.

Semantic rules are written in much the same rule format, except that in a semantic rule, each of the constituents mentioned in the phrase structure skeleton is associated with a logical form. Thus, the semantics for the rule above is

```
sem(whq_ynq_slash_np,
   [([whq,S], s:□),
    (Np, np:□),
    (S, s:[gapsin=np:[gapsem=Np]])]).
```

Here the semantics of the mother `s` is just the semantics of the daughter `s` with the illocutionary force marker `whq` wrapped around it. In addition, the semantics of the `s` gap's `np`'s `gapsem` has been unified with the semantics of the wh-phrase. Through a succession of unifications this will end up assigning the wh-phrase's semantics to the gap position in the argument structure of the `s`. Although each semantic rule must be keyed to a preexisting syntactic rule, there is no assumption of rule-to-rule uniqueness. Any number of semantic rules may be written for a single syntactic rule. We discuss some further details of the semantics in section 2.6

The constituent grammar used in Gemini contains 243 syntactic rules, and 315 semantic rules. Syntactic coverage on the 5875-utterance training set was 94.2%, and on the 756-utterance test set it was 90.9%.

2.4. Parser

Since Gemini was designed with spoken language interpretation in mind, key aspects of the Gemini parser are motivated by the increased needs for robustness and efficiency that characterize spoken language. Gemini uses essentially a pure bottom-up chart parser, with some limited left-context constraints applied to control creation of categories containing syntactic gaps.

Some key properties of the parser are

- The parser is all-paths bottom-up, so that all possible edges admissible by the grammar are found.
- The parser uses subsumption checking to reduce the size of the chart. Essentially, an edge is not added to the chart if it is less general than a preexisting edge, and preexisting edges are removed from the chart if the new edge is more general.
- The parser is *on-line* (Graham, Harrison, and Russo, 1980), essentially meaning that all edges that end at position i are constructed before any that end at position $i+1$. This feature is particularly desirable if the final architecture of the speech understanding system couples Gemini tightly with the speech recognizer, since it guarantees for any partial recognition input that all possible constituents will be built.

An important feature of the parser is the mechanism used to constrain the construction of categories containing syntactic gaps. In earlier work (Moore and Dowding, 1991), we showed that approximately 80% of the edges built in an all-paths bottom-up parser contained gaps, and that it is possible to use prediction in a bottom-up parser only to constrain the gap categories, without requiring prediction for nongapped categories. This limited form of left-context constraint greatly reduces the total number of edges built for a very low overhead. In the 5875-utterance training set, the chart for the average sentence contained 313 edges, but only 23 predictions.

2.5. Typing

The main advantage of typed unification is for grammar development. The type information on features allows the lexicon, grammar, and semantics compilers to provide detailed error analysis regarding the flow of values through the grammar, and to warn if features are assigned improper values, or variables of incompatible types are unified. Since the type-analysis is performed statically at compile time, there is no run-time overhead associated with adding types to the grammar.

The major grammatical category plays a special role in the typing scheme of Gemini. For each category, Gemini makes a set of declarations stipulating its allowable features and the relevant value spaces. Thus, the distinction between the syntactic category of a constituent and its other features can be cashed out as follows: the syntactic category can be thought of as the feature structure

type. The only other types needed by Gemini are the value spaces used by features. Thus for example, the type v (verb) admits a feature **vform**, whose value space **vform-types** can be instantiated with values like present participle, finite, and past participle. Since all recursive features are category-valued, these two kinds of types suffice.

2.6. Interleaving Syntactic and Semantic Information

Sortal Constraints Selectional restrictions are imposed in Gemini through the sorts mechanism. Selectional restrictions include both highly domain-specific information about predicate-argument and very general predicate restrictions. For example, in our application the object of the transitive verb *depart* (as in *flights departing Boston*) is restricted to be an airport or a city, obviously a domain-specific requirement. But the same machinery also restricts a determiner like *all* to take two propositions, and an adjective like *further* to take distances as its measure-specifier (as in *thirty miles further*). In fact, sortal constraints are assigned to every atomic predicate and operator appearing in the logical forms constructed by the semantic rules.

Sorts are located in a conceptual hierarchy and are implemented as Prolog terms such that more general sorts subsume more specific sorts (Mellish, 1988). This allows the subsumption checking and packing in the parser to share structure whenever possible. Semantic coverage with sortal constraints applied was 87.4% on the training set, and on the test set it was 83.7%.

Interleaving Semantics with Parsing In Gemini, syntactic and semantic processing is fully interleaved. Building an edge requires that syntactic constraints be applied, which results in a tree structure, to which semantic rules can be applied, which results in a logical form to which sortal constraints can be applied. Only if the syntactic edge leads to a well-sorted semantically-acceptable logical form fragment is it added to the chart.

Interleaving the syntax and semantics in this way depends on a crucial property of the semantics: a semantic interpretation is available for each syntactic node. This is guaranteed by the semantic rule formalism and by the fact that every lexical item has a semantics associated with it.

Table 2 contains average edge counts and parse timing statistics[1] for the 5875-utterance training set.

[1]Gemini is implemented primarily in Quintus Prolog version 3.1.1. All timing numbers given in this paper were run on a lightly loaded Sun SPARCstation 2 with at least 48 MB of memory. Under normal conditions, Gemini runs in under 12 MB of memory.

	Edges	Time
Syntax only	197	3.4 sec.
Syntax + semantics	234	4.47 sec.
Syntax + semantics + sorts	313	13.5 sec.

Table 2: Average Number of Edges Built by Interleaved Processing

2.7. Utterance Parsing

The constituent parser uses the constituent grammar to build all possible categories bottom-up, independent of location within the string. Thus, the constituent parser does not force any constituent to occur either at the beginning of the utterance, or at the end. Those constraints are stated in what we call the utterance grammar. They are applied after constituent parsing is complete by the *utterance* parser. The utterance grammar specifies ways of combining the categories found by the constituent parser into an analysis of the complete utterance. It is at this point that the system recognizes whether the sentence was a simple complete sentence, an isolated sentence fragment, a run-on sentence, or a sequence of related fragments.

Many systems (Carbonell and Hayes, 1983), (Hobbs et al., 1992), (Seneff, 1992), (Stallard and Bobrow, 1992) have added robustness with a similar postprocessing phase. The approach taken in Gemini differs in that the utterance grammar uses the same syntactic and semantic rule formalism used by the constituent grammar. Thus, the same kinds of logical forms built during constituent parsing are the output of utterance parsing, with the same sortal constraints enforced. For example, an utterance consisting of a sequence of modifier fragments (like *on Tuesday at three o'clock on United*) is interpreted as a conjoined property of a flight, because the only sort of thing in the ATIS domain that can be on Tuesday at three o'clock on United is a flight.

The utterance parser partitions the utterance grammar into equivalence classes and considers each class according to an ordering. Utterance parsing terminates when all constituents satisfying the rules of the current equivalence class are built, unless there are none, in which case the next class is considered. The highest ranked class consists of rules to identify simple complete sentences, the next highest class consists of rules to identify simple isolated sentence fragments, and so on. Thus, the utterance parser allows us to enforce a very coarse form of parse preferences (for example, prefering complete sentences to sentence fragments). These coarse preferences could also be enforced by the parse preference component de-

scribed in section 2.9, but for the sake of efficiency we choose to enforce them here.

The utterance grammar is significantly smaller than the constituent grammar – only 37 syntactic rules and 43 semantic rules.

2.8. Repairs

Grammatical disfluencies occur frequently in spontaneous spoken language. We have implemented a component to detect and correct a large subclass of these disfluencies (called repairs, or self-corrections) where the speaker intends that the meaning of the utterance be gotten by deleting one or more words. Often, the speaker gives clues of their intention by repeating words or adding cue words that signal the repair:

(1) a. How many American airline flights leave Denver on June June tenth.
b. Can you give me information on all the flights from San Francisco no from Pittsburgh to San Francisco on Monday.

The mechanism used in Gemini to detect and correct repairs is currently applied as a fallback if no semantically acceptable interpretation is found for the complete utterance. The mechanism finds sequences of identical or related words, possibly separated by a cue word (for example, oh or no) that might indicate the presence of a repair, and deletes the first occurrence of the matching portion. Since there may be several such sequences of possible repairs in the utterance, the mechanism produces a ranked set of candidate corrected utterances. These candidates are ranked in order of the fewest deleted words. The first candidate that can be given an interpretation is accepted as the intended meaning of the utterance. This approach is presented in detail in (Bear, Dowding, and Shriberg, 1992).

The repair correction mechanism helps increase the syntactic and semantic coverage of Gemini (as reported in Table 1). In the 5875-utterance training set, 178 sentences contained nontrivial repairs[2], of which Gemini found 89 (50%). Of the sentences Gemini corrected, 81 were analyzed correctly (91%), and 8 contained repairs but were corrected wrongly. Similarly, the 756-utterance test set contained 26 repairs, of which Gemini found 11 (42%). Of those 11, 8 were analyzed correctly (77%), and 3 were analyzed incorrectly.

Since Gemini's approach is to extend language analysis to recognize specific patterns characteristic of spoken language, it is important for components like repair correction (which provide the powerful capability of deleting words) not to be applied in circumstances where no repair is present. In the 5875-utterance training set, Gemini misidentified only 15 sentences (0.25%) as containing repairs when they did not. In the 756-utterance test set, only 2 sentences were misidentified as containing repairs (0.26%).

While the repair correction component currently used in Gemini does not make use of acoustic/prosodic information, it is clear that acoustics can contribute meaningful cues to repair. In future work, we hope to improve the performance of our repair correction component by incorporating acoustic/prosodic techniques for repair detection (Bear, Dowding, and Shriberg, 1992) (Nakatani and Hirschberg, 1993) (O'Shaughnessy, 1992).

A central question about the repairs module concerns its role in a tightly integrated system in which the NL component filters speech recognition hypotheses. The open question: should the repairs module be part of the recognizer filter or should it continue to be a post-processing component? The argument for including it in the filter is that without a repairs module, the NL system rejects many sentences with repairs, and will thus dispref essentially correct recognizer hypotheses. The argument against including it is efficiency and the concern that with recognizer errors present, the repair module's precision may suffer: it may attempt to repair sentences with no repair in them. Our current best guess is that recognizer errors are essentially orthogonal to repairs and that a filter including the repairs module will not suffer from precision problems. But we have not yet performed the experiments to decide this.

2.9. Parse Preference Mechanism

In Gemini, parse preferences are enforced when *extracting* syntactically and semantically well-formed parse trees from the chart. In this respect, our approach differs from many other approaches to the problem of parse preferences, which make their preference decisions as parsing progresses, pruning subsequent parsing paths (Frazier and Fodor, 1978), (Hobbs and Bear, 1990), (Marcus 1980). Applying parse preferences requires comparing two subtrees spanning the same portion of the utterance.

The parse preference mechanism begins with a simple strategy to dispref parse trees containing specific "marked" syntax rules. As an example of a dispreferred rule, consider: ***Book those three flights to Boston.*** This sentence has a parse on which *those three* is a noun phrase with a missing head (consider a continuation of the discourse *Three of our clients have sufficient credit*). After penalizing such dispreferred parses, the preference

[2]For these results, we ignored repairs consisting of only an isolate fragment word, or sentence-initial filler words like "yes" and "okay".

mechanism applies attachment heuristics based on the work by Pereira (1985) and Shieber (1983)

Pereira's paper shows how the heuristics of Minimal Attachment and Right Association (Kimball, 1973) can both be implemented using a bottom-up shift-reduce parser.

(2)(a) John sang a song for Mary.
(b) John canceled the room Mary reserved yesterday.

Minimal Attachment selects for the tree with the fewest nodes, so in (2a), the parse that makes *for Mary* a complement of *sings* is preferred. Right Association selects for the tree that incorporates a constituent A into the rightmost possible constituent (where rightmost here means *beginning* the furthest to the right). Thus, in (2b) the parse in which *yesterday* modifies *reserved* is preferred.

The problem with these heuristics is that when they are formulated loosely, as in the previous paragraph, they appear to conflict. In particular, in (2a), Right Association seems to call for the parse that makes *for Mary* a modifier of *song*.

Pereira's goal is to show how a shift-reduce parser can enforce both heuristics without conflict and enforce the desired preferences for examples like (2a) and (2b). He argues that Minimal Attachment and Right Association can be enforced in the desired way by adopting the following heuristics for resolving conflicts:

1. Right Association: In a shift-reduce conflict, prefer shifts to reduces.
2. Minimal Attachment: In a reduce-reduce conflict, prefer longer reduces to shorter reduces.

Since these two principles never apply to the same choice, they never conflict.

For purposes of invoking Pereira's heuristics, the derivation of a parse can be represented as the sequence of S's (Shift) and R's (Reduce) needed to construct the parse's unlabeled bracketing. Consider, for example, the choice between two unlabeled bracketings of (2a):

```
(a)  [John [sang [a song ] [for Mary ] ] ]
      S    S    S S   R S  S     RRR
(b)  [John [sang [ [a song ] [for Mary ] ] ] ]
      S    S     S S   R S  S     RRRR
```

There is a shift for each word and a reduce for each right bracket. Comparison of the two parses consists simply of pairing the moves in the shift-reduce derivation from left to right. Any parse making a shift move that corresponds to a reduce move loses by Right Association. Any parse making a reduce move that corresponds to a longer reduce loses by Minimal Attachment. In derivation (b) above, the third reduce move builds the constituent *a song for Mary* from two constituents, while the corresponding reduce in (a) builds *sang a song for Mary* from three constituents. Parse (b) thus loses by Minimal Attachment.

Questions about the exact nature of parse preferences (and thus about the empirical adequacy of Pereira's proposal) still remain open, but the mechanism sketched does provide plausible results for a number of examples.

2.10. Scoping

The final logical form produced by Gemini is the result of applying a set of quantifier scoping rules to the best interpretation chosen by the parse preference mechanism. The semantic rules build *quasi-logical forms*, which contain complete semantic predicate-argument structure, but do not specify quantifier scoping. The scoping algorithm that we use combines syntactic and semantic information with a set of quantifier scoping preference rules to rank the possible scoped logical forms consistent with the quasi-logical form selected by parse preferences. This algorithm is described in detail in (Moran, 1988).

3. CONCLUSION

In our approach to resolving the tension between overgeneration and robustness in a spoken language understanding system, some aspects of Gemini are specifically oriented towards limiting overgeneration, such as the on-line property for the parser, and fully interleaved syntactic and semantic processing. Other components, such as the fragment and run-on processing provided by the utterance grammar, and the correction of recognizable grammatical repairs, increase the robustness of Gemini. We believe a robust system can still recognize and disprefer utterances containing recognition errors.

Research in the construction of the Gemini system is ongoing to improve Gemini's speed and coverage, as well as to examine deeper integration strategies with speech recognition, and integration of prosodic information into spoken language disambiguation.

REFERENCES

Alshawi, H. (ed) (1992). *The Core Language Engine*, MIT Press, Cambridge.

Bear, J., Dowding, J., and Shriberg, E. (1992). "Integrating Multiple Knowledge Sources for the Detection and Correction of Repairs in Human-Computer Dialog", in *Proceedings of the 30th Annual Meeting of the Association for Computational Linguists*, Newark, DE, pp. 56-63.

Bresnan, J. (ed) (1982). *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge.

Carbonell, J., and Hayes, P. (1983). "Recovery Strategies for Parsing Extragrammatical Language". *American Journal of Computational Linguistics*, Vol. 9, Numbers 3-4, pp. 123-146.

Frazier, L., and Fodor, J.D. (1978). "The Sausage Machine: A New Two-Stage Parsing Model", *Cognition*, Vol. 6, pp. 291-325.

Gazdar, G., Klein, E., Pullum, G., and Sag, I. (1982). *Generalized Phrase Structure Grammar*, Harvard University Press, Cambridge.

Graham, S., Harrison, M., and Ruzzo, W. (1980). "An Improved Context-Free Recognizer", *ACM Transactions on Programming Languages and Systems*, Vol. 2, No. 3, pp. 415-462.

Hobbs, J., and Bear, J. (1990). "Two Principles of Parse Preference", in *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, Vol. 3, pp. 162-167.

Hobbs, J., Appelt, D., Bear, J., Tyson, M., and Magerman, D. (1992). "Robust Processing of Real-World Natural-Language Texts", in *Text Based Intelligent Systems*, ed. P. Jacobs, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 13-33.

Kameyama, M. (1992). "The Syntax and Semantics of the Japanese Language Engine", forthcoming. In *Mazuka, R., and N. Nagai, Eds. Japanese Syntactic Processing*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Kay, M. (1979). "Functional Grammar", in *Proceedings of the 5th Annual Meeting of the Berkeley Linguistics Society*. pp. 142-158.

Kimball, J. (1973). "Seven Principles of Surface Structure Parsing in Natural Language", *Cognition*, Vol. 2, No. 1, pp. 15-47.

MADCOW (1992). "Multi-site Data Collection for a Spoken Language Corpus", in *Proceedings of the DARPA Speech and Natural Language Workshop*, February 23-26, 1992.

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language*, MIT Press, Cambridge.

Moran, D. (1988). "Quantifier Scoping in the SRI Core Language Engine", in *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, State University of New York at Buffalo, Buffalo, NY, pp. 33-40.

Mellish, C. (1988). "Implementing Systemic Classification by Unification". *Computational Linguistics* Vol. 14, pp. 40-51.

Moore, R., and Dowding, J. (1991). "Efficient Bottom-up Parsing", in *Proceedings of the DARPA Speech and Natural Language Workshop*, February 19-22, 1991, pp. 200-203.

Nakatani, C., and Hirschberg, J. (1993). "A Speech-First Model for Repair Detection and Correction", in *Proceedings of the ARPA Workshop on Human Language Technology*, March 21-24, 1993, Plainsboro, NJ.

O'Shaughnessy, D. (1992). "Analysis of False Starts in Spontaneous Speech", in *Proceedings of the 1992 International Conference on Spoken Language Processing*, October 12-16, 1992, Banff, Alberta, Canada, pp. 931-934.

Pereira, F. (1985). "A New Characterization of Attachment Preferences", in *Natural Language Parsing*, Ed. by Dowty, D., Karttunen, L., and Zwicky, A., Cambridge University Press, Cambridge, pp. 307-319.

Pollard, C., and Sag, I. (in press). *Information-Based Syntax and Semantics, Vol. 2*, CSLI Lecture Notes.

Seneff, S. (1992). "A Relaxation Method for Understanding Spontaneous Speech Utterances", in *Proceedings of the Speech and Natural Language Workshop*, Harriman, NY, pp. 299-304.

Shieber, S. (1983). "Sentence Disambiguation by a Shift-Reduce Parsing Technique", in *Proceedings of the 21 Annual Meeting of the Association for Computational Linguistics*, Boston, Massachusetts, pp. 113-118.

Shieber, S., Uszkoreit, H., Pereira, F., Robinson, J., and Tyson, M. (1983). "The Formalism and Implementation of PATR-II", in Grosz, B. and Stickel, M. (eds) *Research on Interactive Acquisition and Use of Knowledge*, SRI International, pp. 39-79.

Stallard, D., and Bobrow, R. (1992). "Fragment Processing in the DELPHI System", in *Proceedings of the Speech and Natural Language Workshop*, Harriman, NY, pp. 305-310.

Uszkoreit, H. (1986). "Categorial Unification Grammars", in *Proceedings of the 11th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Institut fur Kummunikkationsforschung und Phonetik, Bonn University.

Zeevat, H., Klein, E., and Calder, J. (1987). "An Introduction to Unification Categorial Grammar", in Haddock, N., Klein, E., Merrill, G.

(eds.) *Edinburgh Working Papers in Cognitive Science*, Volume 1: *Categorial Grammar, Unification Grammar, and Parsing.*

Focus and Ellipsis in Comparatives and Superlatives: A Case Study

Jean Mark Gawron
SRI International

1. Introduction

The central goal of this paper is to present a semantics of comparatives that deals uniformly with comparative ellipsis and superlatives. Consider (1):

(1) Jean$_i$ gave her$_i$ sister a more expensive book than Alice.

Understandings of the following types are possible:

1. HER SISTER focus: Jean gave Jean's sister a more expensive book than Jean gave Alice.
2. JEAN focus (strict): Jean gave Jean's sister a more expensive book than Alice gave Jean's sister.
3. JEAN focus (sloppy): Jean gave Jean's sister a more expensive book than Alice gave Alice's sister.

In each case, the NP which semantically parallels the NP in the *than*-phrase has been called the focus. I will refer to the NP in the *than*-phrase as the contrast. Now consider the variants in (2), which have analogous interpretations:

(2) Jean gave her sister the most/more expensive book.

1. HER SISTER focus: of all/both x's such that Jean gave x books, Jean gave Jean's sister the most/more expensive book.
2. JEAN focus (strict): of all/both x's such that x gave Jean's sister books, Jean gave Jean's sister the most/more expensive book.
3. JEAN focus (sloppy): of all/both x's such that x gave x's sister books, Jean gave Jean's sister the most/more expensive book.

I will use the term CONTRAST-SET to describe the set of entities whose properties are being measured and compared, a set which always includes the denotation of the focus. In the paraphrases above, the contrast-set is described by the *of*-phrase. I will call the nonelliptical focus constructions in (2) maximal-degree constructions (rather than superlative constructions) because they come with both comparative and superlative morphology. The only difference between the two is whether or not the contrast-set is presupposed to have two members.

Each of the three readings in (2) can be obtained from the corresponding reading of (1) simply by quantifying over the argument position filled by the contrast. Sentence (2) has another reading with no parallel in (1). This is the reading on which no givings are presupposed. There is simply a set of books available in the discourse, and Jean has given her sister the most expensive. I will refer to the minimal NP containing the comparative element as the COMPARATIVE NP in comparatives and the SUPERLATIVE NP in superlatives. For this reading, I will say that the superlative NP is the focus. One kind of elliptical comparative which makes a parallel comparison is shown in

(3) Jean gave her sister a more expensive book than *War and Peace.*

Here, too, only one giving event is at issue. What is being compared is the expense of the book in that giving event with the expense of *War and Peace.*

The basic conclusion I draw from (1), (2), and (3) is the following: for both constructions interpretations vary according to which NP is taken as focus. In effect, the same interpretive difficulties that arise in comparatives arise in maximal-degree constructions.

I will argue below that there is a striking similarity between the pattern of readings in (1) and (3) and a pattern typical of the interaction of focus and quantification. Consider, two different focus possibilities for (4):

(4) a. Most New Yorkers eat Chinese food with CHOPSTICKS.
b. Most New Yorkers eat CHINESE FOOD with chopsticks.

The two focus possibilities correspond roughly to the following readings:

(5) a. Most New Yorkers who eat Chinese food with something eat Chinese food with CHOPSTICKS.
b. Most New Yorkers who eat something with chopsticks eat CHINESE FOOD with chopsticks.

In each case the focus construction can be thought of as adding a restriction to the quantification. The restriction is obtained by abstracting the focus out of the main clause semantics and existentially quantifying it away. I will follow Jacobs 1991 by calling the property obtained by abstracting the focus out of the main clause semantics the BACKGROUND.

Consistent with a number of other analyses (beginning with Cresswell 1976), this treatment will interpret both comparatives and superlatives as a quantification over degrees; the various readings above are all obtained by restricting the comparative quantification with different backgrounds.

As remarked above, (2) has both superlative and comparative variants. Thus, comparative morphology is compatible with maximal-degree semantics. Some sentences are ambiguous. Consider:

(6) Who's taller?

Sentence (6) might be uttered in two different sorts of contexts:

(7) a. Their center is not the tallest member of the team. Who's taller?
b. John and Bill weigh the same. Who's taller?

In (a), the question is which member of the team under discussion is taller than the center. This is a discourse-bound comparative. In (b), the discourse provides a contrast-set and the question is who in that set has the maximum height. Since the set has cardinality two, the comparative form of the adjective is licensed. The second sentence in (b) might be replaced with any of the following:

(8) a. Of the two, who's taller?
b. Who's taller, John or Bill?
c. Is John or Bill taller?

All of these unambiguously call for a maximal-degree interpretation.

The comparative construction exhibits a bewildering range of elliptical phenomena. This paper is concerned with COMPARATIVE ELLIPSIS. I take it that all of the following are elliptical:

(9) a. John has met more presidents than Mary.
b. John has met more presidents than Mary has.
c. John has met more presidents than Mary has met.
d. John owns pictures of more presidents than Mary owns.
e. John owns more trucks than Mary does cars.

Sentence (9a) illustrates what I will call comparative ellipsis; (9b) illustrates the comparative construction interacting with verb-phrase ellipsis; (9c) illustrates the almost obligatory deletion of the head noun of the degree NP in the *than*-clause when it is identical with the head noun of the comparative NP; and (9d) illustrates what may be a more extreme version of the same thing. Sentence (9e) illustrates gapping in a comparative clause. Dealing with all these examples would be well beyond the scope of this paper.

Having stated the practical agenda for the paper, I will add that I do not foresee any problems of principle. The approach to both ellipsis and focus that I will adopt is from Dalrymple, Shieber, and Pereira 1991 (henceforth DSP), a paper which deals primarily with verb-phrase ellipsis.[1] The DSP framework shows promise of being a very general tool with which to approach phenomena of ellipsis. It seems likely that examples of the type exhibited in (9b) and (9e)

[1]Pulman 1991 also proposes applying the DSP framework to comparative ellipsis. The details of the analysis are different, but the approach is very much in the spirit of what is argued here.

do not present problems particular to comparatives. Sentences (9c) and (9d) do raise issues particular to comparatives, but the form of ellipsis shown there is largely orthogonal to the central issues of this paper. I emphasize sentences like (9a) because these are the examples that behave most like other focus constructions with regard to the scope-of-focus issues discussed in Section 2.1.

I will distinguish between degree and quantity comparatives. Degree comparatives are adjectival or adverbial. Quantity comparatives involve number or amount:

Degree: John drove faster than Mary.
John was taller than Mary.
Quantity: John ate more apples than Mary.
John drank more wine than Mary.

Due to limitations of space, I will deal only with degree comparatives in this paper. There are some interesting issues involved in extending the account here to quantity comparatives, which show somewhat different ranges of readings of scope properties. For a fuller discussion, see Gawron 1992.

2. Parallels between Measure Constructions and *Only*

The primary point of this section is to draw parallels between comparative ellipsis and other focus constructions. It is clear from the examples discussed in Section 1 that the *than*-phrase in comparative ellipsis seeks to associate with a focus much as a word like *only* does. Thus, interpreting elliptical comparatives and superlatives entails determining a focus or foci and a scope of focus.

2.1. Scope of Ellipsis and Scope-Fixing

Consider first the ambiguity of a sentence like:

(10) John wants to own more records than Mary.

Sentence (10) can be paraphrased with either (11a) or (11b):

(11) a. Wide scope: John wants to own more records than Mary wants to own.
b. Narrow scope: John wants to own more records than Mary owns.

In the wide-scope reading, the comparison is between desires; in the narrow-scope reading, the comparison is between the number of records John owns and the number John owns, and John wants that comparison to work out a certain way.[2] As the paraphrases suggest, there is an ambiguity in how much missing material has to be reconstructed. Now consider a superlative example:

[2]Paraphrase (b) here actually collapses two distinct *de re* and *de dicto* readings, but that does not affect the point under discussion.

(12) John wants to own the most records.

Again, two readings are possible:

(13) a. John wants to own more records than anyone else wants to own.
b. John wants to own more records than anyone else owns.

There is a difference between (11) and (13) in these cases; the attachment of the *than*-phrase gives the comparative construction a syntactic way of *fixing* the scope of ellipsis. Consider the following:

(14) John wants to own more records than Mary by next year.

Sentence (14) has only a narrow-scope reading: what John wants is that by next year his collection is bigger than Mary's. A natural explanation is that the modifier *by November* most naturally attaches low, thus forcing low attachment of the *than*-phrase. Low attachment of the *than*-phrase means narrow scope-of-focus.

In light of this evidence, we propose Hypothesis A, to be revised later:

Hypothesis A

The sister of *than*-phrase is the scope-of-focus in comparative ellipsis.

The simple picture of comparative ellipsis is this: there is a relation between an individual and a measure and the measure-values of the relation are compared for the focus and the contrast. By the scope-of-focus in Hypothesis A, I mean the constituent whose semantics provides the relation being compared. In the wide-scope reading of (10), that constituent is the VP *wants to own more records.* In the narrow-scope reading, that constituent is the VP *own more records.*

In being governed by something like Hypothesis A, comparative ellipsis sentences with *than* resemble sentences with *only.* Scope-fixing effects with *only* are discussed in Taglicht 1984 and Rooth 1985:

(15) a. They were advised to only learn Spanish.
b. They were only advised to learn Spanish.

Here (a) has the reading on which advice is given to ignore languages other than Spanish; (b) has the reading on which the only advice given was to learn Spanish. The (a) sentence lacks the reading available for the (b) sentence, and vice versa. Thus, syntactic attachment of *only* fixes the scope of ellipsis, just as the syntactic attachment of the *than*-phrase does. The sentences in (15) are unambiguous only by a syntactic accident. The word *only* attaches verb-phrase initially so that it is clear which verb-phrase it has chosen; the

than-phrase attaches verb-phrase finally, so that sentences like those in (13) may be ambiguous.

2.2. Entailments in Adjectival Comparatives

Noun phrases analogous to the following are noted in Bresnan 1973:

(16) a. A stronger man than John was found.
b. ?A stronger man than Mary was found.
c. A man stronger than John was found.
d. A man stronger than Mary was found.

One would like these facts to fall out from Hypothesis A. That is, all of the NPs in (16) are elliptical, and what they are elliptical for is determined by how much material is C-commanded by the *than*-phrase. Thus, one's account of ellipsis, guided by Hypothesis A, ought to give the NPs semantics roughly like the following:

(17) a. An m strong man such that [$m > s$ and John is an s strong man]
b.?An m strong man such that[$m > s$ and Mary is an s strong man]
c. A man m strong such that [$m > s$ and John is s strong]
d. A man m strong such that [$m > s$ and Mary is s strong]

An interesting property of these cases is that they appear related to some exceptions to Hypothesis A (discussed in Section 2.1). Consider:

(18) a. A more competent engineer than Bonnie was hired.

An m competent engineer such that [$m > s$ and Bonnie is an s competent engineer] was hired.

b. A more competent engineer was hired than Bonnie.

An m competent engineer was hired such that [$m > s$ and Bonnie, an s competent engineer, was hired].

A literal application of Hypothesis A would lead one to expect that these had something like the indicated paraphrases, but in fact sentences (a) and (b) do not appear to differ on their possible readings. Crucially, (b) has no entailment that Bonnie was hired. Contrast the sort of case which motivated Hypothesis A:

(19) BONNIE hired a more competent engineer than Frieda.

Here, if Bonnie is being compared to Frieda (that is, if *Bonnie* is the focus), then Frieda has to have hired a engineer.

We can sum up the facts from this section and Section 2.1 with the following observation:

Observation

(a) When the comparative NP is the focus, the syntactic scope-of-focus is the comparative N-bar.
(b) Otherwise the syntactic scope-of-focus is the surface sister of the *than*-phrase.

One might eliminate the disjunctive nature of this observation in either of two ways. First, one might assimilate (18b) to extraposition, and apply Hypothesis A only to the source. The drawback of this approach, it seems to me, is that it offers no explanation of the facts. Although an extraposition analysis will capture the actual reading of (18), it gives no account of why other readings aren't possible. To correctly constrain the readings, we will need to restrict *than*-phrases to N-bar attachment when the focus is the comparative NP. But this restrictions will be lifted when the focus is anything else. The other way to go is to look for a semantic explanation. This is what I will propose below.

3. Semantics of Comparatives

3.1. Subdeletion

To illustrate the approach to the semantics of comparatives taken here, it will be useful to start with a noncomparative example:

(20) This desk is six feet wide.

I will represent the semantics of degree adjectives as a relation between individuals and degrees:

(21) wide (that-table, [foot 6])

The term [foot 6] denotes a measure in an ordered set of measures with the sort of structure discussed in Krifka 1987 and Nerbonne 1991. It is not crucial to the issues discussed in this paper that degree adjectives be relations between individuals and degrees, but it is crucial that the semantics of a simple measure assertion like (21) have in it terms that correspond to an individual being measured and a measure.

I will also assume that adjectival relations are downwardly monotonic on their measure arguments, so that if (21) is true then

(22) wide (that-table, [foot 5])

is also true. So the truth-conditions of (21) will only require that table to be at least 6 feet wide. One advantage of this downward monotonicity is that the semantics of *that table is wide* can just be:

(23) wide (that-table, STANDARD)

where STANDARD is some pragmatically fixed standard. The truth-conditions of (23) will then require that table to be at least as wide as the standard.

The kind of comparative that is easiest to understand semantically occurs relatively infrequently:

(24) This desk is longer than that table is wide.

I assume that (25) provides a satisfactory logical representation of (24):

(25) $\forall ?s$ [wide(that-table, $?s$),
$\exists ?m$ [$>$($?m$, $?s$),
long(this-desk, $?m$)]]

Glossing the semantics: every degree s that is in the width relation to that table is such that there exists a degree m greater than s that stands in the length relation to this desk.

One reason for the universal quantification is the downward monotonicity of the adjective relation. We need to require this desk to have a length taller than all the widths of that table in order to be sure that the maximal width is included. There are other motivations for the universal quantification, however. One is that the *than*-phrase is a negative polarity context:

(26) John is smarter than any bureaucrat.

Another is the behavior of comparatives in modal contexts:

(27) John can run faster than Bill.

This sentence should come out true only if John can run faster than any speed Bill can run. To get this right, one would need universal quantification even if the adjective relations weren't downwardly monotonic.[3]

The central claim of this semantics is that the comparative construction introduces a quantifier on measures restricted by the material in the *than* phrase.[4]

I will assume that each measure set has an ordering relation on measures which I will notate simply as $>$, and that comparatives use $>$. I will call

[3]Thanks to Bob Moore for pointing this example out.

[4]I will refer to the second-order property obtained by abstracting on ψ in:

$\forall s$ [$\phi(s)$,
$\exists m[>(m, s), \psi(m)]]$

as the comparative quantifier; thus, ψ stands as the comparative quantifier's scope. Of course, there are really two quantifiers here, and they can scope independently, but for most of the examples under consideration that possibility is not germane to the discussion. This paper has little to say about constraints on the scoping possibilities of the comparative quantifier.

the measure constrained by the main clause the STANDARD and the measure constrained by the *than*-clause the REFERENCE.

3.1. Comparative Ellipsis

We now turn to cases involving ellipsis. We begin with a brief summary of the framework of DSP, using a verb phrase ellipsis example:

(28) a. Bill washed his car and John did too.
b. AND[wash(b,car(b)), P(j)]

Given the semantics in (b), the problem of interpreting (a) now reduces to the problem of solving for the unspecified property P. In DSP, resolving that property involves the following steps.

1. Locate source: wash(b,car(b)).

2. Establish parallel elements and locate *primary occurrences* in source.

   $$\text{wash}\,(\underline{\text{b}}, \text{car(b)})$$

   Parallel elements are constituents in a tree. Primary occurrences are terms in the semantic form. A primary occurrence in the source is a term actually contributed by a parallel element. Thus, the two subjects are parallel in (28a), and the first occurrence of *b* above is primary because it is contributed by the subject NP in the source. The second is not because it is contributed by a pronoun which is not a parallel element.

3. Set up equation.

   $$P(\text{b}) = \text{wash}\,(\underline{\text{b}}, \text{car(b)})$$

4. Solve equation.

   $$\begin{array}{ll} \text{Strict:} & P = \lambda x[\text{wash}\,(x, \text{car(b)})] \\ \text{Sloppy:} & P = \lambda x[\text{wash}\,(x, \text{car}(x))] \\ & P = \lambda x[\text{wash}\,(\underline{\text{b}}, \text{car}(x))] \\ & P = \lambda x[\text{wash}\,(\underline{\text{b}}, \text{car(b)})] \end{array}$$

5. Discard UNACCEPTABLE SOLUTIONS, that is, solutions which contain a primary occurrence. DSP reject certain solutions that violate parallelism in that they do not abstract over a primary occurrence. In this case the single primary occurrence is the occurrence of **b** filling the first argument role of *wash*. Thus, the third and fourth solutions above are unacceptable.

We now turn to cases of comparative ellipsis:

(29) Jean gave her sister a more expensive book than Alice.

The semantics is

(30) $\exists y$ [$\forall s$ [R(a, s),
$\exists m$ [$>$(m, s),
AND[book(y),
expensive(y, m)]]],
give(j , sister(j) , y)]

The idea here is that what the *than*-phrase contributes is just a relation between an individual and a measure:

R(a, s)

Note that is not meant to commit the syntax in any way to an empty measure element.

On the approach to the semantics of comparatives we have adopted, the *than*-phrase always introduces a proposition which restricts the comparative quantifier, whether or not the sentence is elliptical. In the elliptical sentences all we have restricting the quantifier is an unspecified relation between an individual and a degree. The problem of interpreting the elliptical sentences now reduces to the problem of resolving the relation R. We will resolve the relation by abtracting elements out of the semantics of the main clause. Thus we have a paradigm case of the interaction of focus and quantification as discussed in section 1. A relation is being contributed by the semantics of the main clause (this is what corresponds to the background of Jacobs 1991), and that relation restricts the domain of quantification.

In the framework of DSP, solving for R means setting up a second-order equation on the basis of parallelisms between the elliptical semantics and some template semantics. The steps are as follows:

1. Locate scope-of-focus. We will use the term scope-of-focus rather than source because, as illustrated in section 2.1, there are ambiguities in comparative ellipsis that can be captured only if the amount of material omitted in the ellipsis is allowed to vary. In this case, the template on which the elliptical clause will be built is just the semantics of the main clause minus the comparative quantifier. That the comparative quantifier must always be abstracted out before setting up equations is just a stipulation about degree constructions (the account of maximal-degree constructions will entail the same move):

(31) $\exists y$[AND[book(y),
expensive(y, m)],
give(j , sister(j) , y)]]

2. Establish parallel elements and locate primary occurrences in source. In comparative ellipsis, there are two parallelisms to worry about. One will be established simply by locating parallel elements in a syntactic tree. This is the parallelism of the focus and contrast. The other parallelism is that between the standard measure and the reference measure. Not wishing to adopt an abstract syntactic analysis for these cases, I will simply assume that parallelism of degrees is given by the construction. Thus, the unique occurrence of the standard in (31) will be a primary occurrence. Let us consider the case where *Jean* is focus.

Main Clause:	JEAN gave her sister an	m expensive book
	Focus	Standard
Than Clause:	Alice	s
	Contrast	Reference

3. Set up and solve equations.

(32) JEAN as focus: $R(\mathrm{j}, m) = \exists\, y[\textsc{and}[\mathrm{book}(y), \mathrm{expensive}(y, \underline{m})], \mathrm{give}(\underline{\mathrm{j}}, \mathrm{sister}(\mathrm{j}), y)]$

Strict: $R = \lambda x, z\, [\exists\, y[\textsc{and}[\mathrm{book}(y), \mathrm{expensive}(y, z)], \mathrm{give}(x, \mathrm{sister}(\mathrm{j}), y)]]$

Sloppy: $R = \lambda x, z\, [\exists\, y[\textsc{and}[\mathrm{book}(y), \mathrm{expensive}(y, z)], \mathrm{give}(x, \mathrm{sister}(x), y)]]$

Substituting the acceptable solutions for R in (30) yields the desired result.

4. Discard unacceptable solutions. Again these are just the solutions that have primary occurrences in them. There are five unacceptable solutions in all, two which fail only in leaving behind the primary occurrence of the focus, two which fail in leaving behind both primary occurrences, and one which fails in leaving behind the primary occurrence of the standard. Here are two of them:

(33) $R = \lambda x, z\, \exists\, y[\textsc{and}[\mathrm{book}(y), \mathrm{expensive}(y, z)], \mathrm{give}(\underline{\mathrm{j}}, \mathrm{sister}(x), y)]]$

(34) $R = \lambda x, w\, \exists\, y[\textsc{and}[\mathrm{book}(y), \mathrm{expensive}(y, z)], \mathrm{give}(\underline{\mathrm{j}}, \mathrm{sister}(x), y)]]$

The first of these would give the impossible reading: *Jean gave Jean's sister a more expensive book than Jean gave Alice's sister.* The second is just vacuous abstraction on both argument positions and would give the contradictory reading that Jean gave her sister a more expensive book than Jean gave her sister. The reader may verify that the other three unacceptable solutions all give impossible readings.

The other reading to deal with is the case where *her sister* is the focus. In this case the equation is:

(35) HER SISTER: $R(\text{sister(j)}, m) = \exists\, y[\text{AND}[\text{book}(y),$
$\text{expensive}(y, m)],$
$\text{give}(\text{j}, \underline{\text{sister(j)}}, y)]$
$R = \lambda x, z\,[\exists\, y[\text{AND}[\text{book}(y),$
$\text{expensive}(y, z)],$
$\text{give}(\text{j}, x, y)]]$

In this case there is only one acceptable solution because there is only one primary occurrence for each argument of the relation. There are three unacceptable solutions, one which leaves behind just the primary occurrence of the focus, one which leaves behind just the primary occurrence of the standard, and one with vacuous abstraction on both argument positions of R, which leaves behind both.

We turn now to the other example of comparative ellipsis discussed in Section 1:

(36) Jean gave her sister a more expensive book than *War and Peace.*

The semantics is:

(37) $\exists\, y\ [\forall\, s\ [\text{R}(\text{War-and-Peace}, s),$
$\exists\, m\ [>(m, s),$
$\text{AND}[\ \text{book}(y),$
$\text{expensive}(y, m)]]],$
$\text{give}(\text{j}, \text{sister(j)}, y)]$

The equations for this scope-of-focus are:

(38) $R(y, m) = \text{AND}[\ \text{book}(y),$
$\text{expensive}(y, m)]$
$R = \lambda x, z\,[\text{AND}[\ \text{book}(x)$
$\text{expensive}(x, z)]]$

Since R is applied to *War and Peace*, the sentence will be true only if *War and Peace* is a book. This, then, is one step in accounting for the entailment facts

noted in Bresnan 1973 and discussed in Section 2.2. We still need to explain why this is the correct scope-of-focus for those examples, however.

In this case the head noun and the adjective predications must both contain primary occurrences. Among the unacceptable solutions, there are two ruled out simply because they do not abstract over one of the two primary occurrences of y:

(39) $R = \lambda x, z$ [AND[book(y),
expensive(x, z)]]
$R = \lambda x, z$ [AND[book(x)
expensive(y, z)]]

The first reading would not preserve the entailment that *War and Peace* is a book (see Section 2.2). The second would contradictorily require that y be more expensive than itself.

In calling both occurrences of y primary occurrences here, we are building on the sense of primary occurrence as it is assumed in DSP. The motivation for this move is the following: the two occurrences of y in the equations in (39) differ from the two occurrences of j in (32) in that the grammar always requires the two occurrences of y to be identified. An adjective modifying a noun always has its theme argument identified with the noun's. One may think of the semantics of the N-bar as being:

[book $\wedge$ λx[expensive(x, z)]](w)

Here $\wedge$ represents property conjunction. From this perspective there is really only one primary occurrence of the N-bar variable. What is going on here is reminiscent of other cases where the grammar requires identification of two variables, such as the cases of obligatorily sloppy pronouns in Serbo-Croatian discussed in DSP. A more familiar case would be the cases of obligatory sloppy readings with raising verbs such as *expect* in

(40) John expects to leave and Bill does too.

Here there is no reading on which Bill expects John to leave. Yet there is good motivation for believing that *expect* takes a proposition argument, and that the semantics of the source clause is

(41) expect(j,leave(j))

Blocking the strict reading would entail hypothesizing two primary occurrences.

We have now worked through the semantics of two closely related elliptical examples, arguing that the principal difference between them is a difference in the scope-of-ellipsis. It should be clear from these examples that any hopes

this analysis may have in being explanatory lie in being able to give a principled account of how the scope-of-focus is determined. Consider again the semantics shown in (30). What would have happened if we had chosen the scope-of-focus in (31) with the comparative NP as the focus? The reading predicted then would have been incorrect:

(42) Jean gave her sister an m expensive book and Jean gave her sister *War and Peace*, an s expensive book, and m was bigger than s.

This is essentially the same fact we noted for (18).

I will now argue that for semantic reasons the maximal scope-of-focus when the comparative NP is focus is the N-bar. Consider (37). There are four cases to look at:

1. Nbar scope: okay.

2. The scope-of-focus is the scope of the indefinite.

$$R(y, m) = \text{give}(\text{j}, \text{sister}(\text{j}), y),$$

   Here there is no occurrence of m on the right-hand side of the equation. Therefore, this equation has no solution that does not involve vacuous abstraction.

3. The scope-of-focus is the sentence with indefinite quantified in and r is a first-order relation. The equation then is

$$\begin{aligned} R(y, m) = \exists\, y[\text{AND}[\,&\text{book}(\,y), \\ &\text{expensive}(\,y,\, m)], \\ &\text{give}(\,\text{j}\,,\ \text{sister}(\text{j})\,,\ y)]] \end{aligned}$$

   The problem with this equation is that there is no occurrence of y, the focus, on the right-hand side. Since the quantifier has been quantified in, any y on the right hand side is a bound variable and no solution can abstract over it. Again, the equation has no solutions which do not involve vacuous abstraction.

4. The scope-of-focus is the sentence with indefinite quantified. R is a higher-order relation. The system in DSP allows type-lifting in order to deal with cases where one or both of the parallel elements is a quantifier. Thus, in analyzing:

   Every student revised his paper, and John did too.

John can be made parallel to *Every student* by type-lifting. On this account (36), *War and Peace* is parallel not to an individual-level variable, but to the indefinite quantifier, *a more expensive book.* It is thus type-lifted to be a quantifier:

$$\lambda P[P(\text{War-and-Peace})]$$

and R is correspondingly type-lifted to allow a quantifier to be one of its arguments. The resulting equation is

$$\mathrm{R}\left(\begin{array}{l}\lambda P[\exists\, y\,[\textsc{and}[\ \text{book}(y),\\ \qquad\qquad\qquad \text{expensive}(y,\ m)]\\ \qquad\quad P(y)]\,]\end{array}\ ,m\right) = \exists y\left[\begin{array}{l}\textsc{and}[\,\text{book}(y),\\ \qquad\quad \text{expensive}(y,\ m)],\\ \text{give}(\ \text{j}\ ,\ \text{sister(j)}\ ,\ y)\end{array}\right]$$

But this, too, has no solutions which do not involve vacuous abstraction. In this case no solution can simultaneously abstract over the focus quantifier and m the standard. Two of the solutions are

$$\mathrm{R} = \lambda\mathcal{P}, z[\mathcal{P}\,(\lambda y[\text{give}(\ \text{j}\ ,\ \text{sister(j)}\ ,\ y)]\,)]$$
$$\mathrm{R} = \lambda\mathcal{P}, z[\exists\, y[\textsc{and}[\,\text{book}(y),\ \text{expensive}(y,\ z)],\ \text{give}(\ \text{j}\ ,\ \text{sister(j)}\ ,\ y)]\,]$$

There is also a solution which vacuously abstracts over both argument positions.

If we could eliminate all the equations that have only vacuous solutions, then we would have an account of why the N-bar is the only scope-of-focus in this case. Careful readers of DSP will note that they posit no restriction against vacuous solutions. Instead, unacceptable solutions are characterized as those which still contain a primary occurrence. This rules out many cases of vacuous abstraction, but it also rules out solutions such as (33). Rather than try to modify this characterization, I want to suggest that there is an independent restriction, not on solutions, but on equations, which rules out those that have no nonvacuous solutions. This restriction should be thought of as an adjunct to the algorithm for finding a source and parallel elements and setting up an equation. An equation which has no nonvacuous solutions is simply one for which no true parallelisms have been found.

We can now revise Hypothesis A of Section 2.1 and propose a semantic account of the scope-of-focus facts observed in (18):

Hypothesis A: Final Version

The syntactic scope-of-focus is the maximal constituent of the surface sister of the *than*-phrase whose semantics can provide a scope-of-focus with acceptable ellipsis equations.

Note that with this hypothesis, we have an account of the adjectival entailment facts noted in Bresnan 1973 and discussed in section 2.2

(43) ? A stronger man than Mary was found.

The widest scope-of-focus that yields an acceptable equation is the N-bar. There is one narrower scope-of-focus than that N-bar that yields equations with acceptable solutions, namely, the semantics of the adjective:

(44) strong(y, m)

But Hypothesis A, on syntactic grounds, rules out choosing this as the scope-of-focus for (43). It follows from this that any equations resolving the ellipsis will have to include the noun predication in their solutions for R. Thus, any solutions will entail that Mary is a man.

3.2. Maximal-Degree Consturctions

We begin by presenting the semantics for (2), reproduced here:

(45) Jean gave her sister the most expensive book.

The semantics, irrespective of what the focus i,s is

(46) the y [$\forall\, s$ [$\exists x$[C(x), R(x, s)],
$\exists\, m$ [$\geq$(m, s),
AND[book(y),
expensive(y, m)]]],
give(j , sister(j) , y)]

There are several differences here from the semantics of a comparative ellipsis sentence. First, the position filled by the contrast in the *than*-phrase has been existentially quantified over, with that quantification restricted to the members of a contrast-set C. Under the scope of $\forall$, this has the effect of a universal quantification. Second, the ordering relation has been changed from $>$ to $\geq$. This is because the focus is in the contrast-set too, and if the sentence is ever to be uttered truthfully, ties with the highest scoring element of the contrast set must be allowed.[5]

One might argue for the inclusion of the contrast-set C in (46) on the basis of a general requirement that all quantification should be contextually restricted. But independently of that there is a specific motivation for making it explicit in the semantics of superlatives. Sometimes the contrast-set can be associated with syntactically overt material:

[5]The only difference in the semantics of *Jean gave her sister the more expensive book* is that instead of quantifying over the contrast-set with $\exists$ we quantify with $(\exists; 2)$.

(47) a. Of the three sisters, Jean bought the most expensive book.
b. Which sister bought the most expensive book?

Thus, (47a) is appropriate only when JEAN is the focus, and the set of buyers Jean will be compared to is the set of the three sisters in question, which must include Jean. In (47b), on what is probably the most accessible reading, the contrast-set is identified with the restriction-set of the *wh*-phrase.

The equations for the case when *Jean* is focus and for the case when *her sister* is focus are exactly as they were for the comparative analogue discussed in Section 3.2, as are the solutions. As was noted in Section 1.1, sentence (46) has another focus possibility, parallel not to (29) but to (36). In this case the focus is the superlative NP. The equation for this reading is exactly the same as the equation for (36), given in (38).

Another difference between the superlatives and the comparatives is that no version of Hypothesis A applies to the superlatives, since they have no *than*-phrase. Thus, nothing prevents a reading in which the scope of focus is narrower than N-bar when the focus is the superlative NP:

(48) Of the three items the clerk showed, Jean bought the most expensive ring.

Here the items need not be all rings. The scope-of-focus must be the adjective-phrase alone:[6]

4. Conclusion

In this paper I have proposed an analysis of measure constructions that provides a uniform semantics for comparative ellipsis and superlatives, arguing that both can be regarded as examples of focus constructions. The specialness of comparatives ellipsis consists in requiring a contrast along with a focus.

The analysis proposes an account of the entailments of degree comparatives in which the comparative NP is the focus. Thus,

(49) A stronger man than Bill was found.

entails that Bill was a man. This is accounted for by the relationship between the scope-of-focus and the *than*-phrase.

I conclude with an effort to show that the equational machinery of DSP does extend neatly to handle a paradigm case of a focus construction. The following is a reworking of the analysis of *only* in Rooth 1985:

(50) John only introduced Sue to her brother.

[6]Thanks to Carl Pollard for pointing this reading out.

(51) $\forall p[\,\exists\, x\,[\,\mathrm{A}(x), \mathrm{AND}[\check{}p, (\mathrm{P}(x) = p)]\,],$
$(p =$ introduce(j, brother(s), s) $)\,]$

SUE: $P(\mathrm{s}) =$ [introduce(j, brother(s), s)]
$P = \lambda y$ [introduce(j, brother(y), y)]
$P = \lambda y$ [introduce(j, brother(s), y)]

HER BROTHER: P(brother(s)) = [introduce(j, brother(s), s)]
$P = \lambda y$ [introduce(j, y, s)]

The resemblance of the proposed semantics to the semantics of maximal measure constructions is striking. Instead of a universal quantification over measures, there is a universal quantification over propositions. Most interestingly, in both cases, the restriction of the universal requires an existential quantification over a pragmatically given set. In the case of comparatives, I have called that the contrast-set; Rooth calls A the alternative-set, characterizing the members of A as the alternatives to the focus in the discourse. In the case where *her brother* was focus, Rooth 1985 would associate two things with (50):

(52) a. $\forall p[\mathrm{C}(p) \wedge \check{}p \longrightarrow p =$ introduce(j, brother(s), s)]
b. $\lambda p \exists y[[\mathrm{A}(y)] \wedge p =$ introduce(j, y, s)]

The first is roughly the semantics of the sentence, independent of what the focus is; the second is the p-set (or presupposition set) that goes with having *her brother* as focus. The p-set property in (52b) is then identified with the property of propositions C in (52a). In the recasting given in (51) predicating C of p has been replaced by predicating property P of any individual x and requiring proposition p to be equal to the resulting proposition. The equations solving for P are then set up depending on what has been chosen as the focus. In effect, the task of recursively building up p-sets in parallel with the main semantics is being taken over by the equation-solving machinery. Rooth's idea that one component of the semantics should be kept independent of what the focus is has been preserved. In fact, that property has been preserved throughout this paper: the semantics independently of a solved equation is always compatible with any focus in the scope-of-focus.

Rooth's approach shares with that of Jacobs 1991 the idea that an account of focus requires recourse to some two-component account of meaning. In Rooth it is the main translation and the p-set; in Jacobs it is the focus and the background. One interesting feature of the equational approach is that it tries to make do with a single meaning component, which can then generate a variety of restrictions on the quantifications of focus operators.

Acknowledgment

I owe the basic skeleton over which this paper has been hung to Carl Pollard's unpublished work on the syntax and semantics of comparatives, and much of my thinking on the semantics of measure and quantification with measures to discussions with John Nerbonne. The analysis of comparatives implemented in the HPNL system at Hewlett-Packard Labs was the combined effort of a number of people, all of whom at various times help me understand what was going on. They included Dan Flickinger, David Goddeau, Masayo Iida, Bill Ladusaw, and Lyn Walker. Discussions with Mary Dalrymple have had an obvious effect on my views on ellipsis, as have more recent discussions with her co-authors Stuart Shieber and Fernando Pereira. More recently, collaborating with Bob Moore at SRI on an implementation of an analysis of comparatives in the Gemini system, has changed my views on the comparative quantifier.

This research was supported by the Defense Advanced Research Projects Agency under Contract ONR N00014-90-C-0085 with the Office of Naval Research. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency of the U.S. Government.

References

Barwise, J. and Cooper, R. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4(2):159-219.

Bresnan, J. 1973. Syntax of the Comparative Clause Construction in English. *Linguistic Inquiry* 4(2):275-343.

Cresswell, M.J. 1976. The Semantics of Degree. In B. Partee (ed). *Montague Grammar.* Academic Press, New York.

Dalrymple, M.,Shieber,S., and Pereira,F. 1991. Ellipsis and Higher-Order Unification. *Linguistics and Philosophy* 14:399-452.

Evans, G. 1977. Pronouns, Quantifiers, and Relative Clauses. *Canadian Journal of Philosophy* 7:467-536.

Gawron, J.M. 1992. Comparatives, Superlatives, and Focus. Manuscript. SRI, International.

Geach,P. 1962. *Reference and Generality.* Cornell University Press, Ithaca.

Jacobs, J. 1991. Focus Ambiguities. *Journal of Semantics* 8: 1-16.

Klein, Ewan. 1980. The Interpretation of Adjectival Comparatives. *Journal of Linguistics* 18:1-45.

Krifka, M. 1987. Nominal Reference and Temporal Constitution: Towards a Semantics of Quantity. Technical Report 17, Forschungsstele fur natursprachliche Systeme, Universitat Turbingen.

Link, G. 1983. The Logical Analysis of Plurals and Mass Terms: A Lattice-Theoretical Approach. In R. Bauerle, U. Egli, and A. von Stechow (eds).

Meaning, Use and the Interpretation of Language. de Gruyter, Berlin.
Nerbonne, J. 1991. Nominal Comparatives and Generalized Quantifiers. Manuscript. DFKI, Saarbrucken.
Pulman, S. 1991. Comparatives and Ellipsis. Proceedings of European ACL.
Rayner, M. and A. Banks. 1990. An Implementable Semantics for Comparative Constructions. *Computational Linguistics* 16: 86-112.
Rooth, M. 1985. Association with Focus. Ph.D. Dissertation, University of Massachusetts at Amherst.
Sag, I. *Deletion and Logical Form.* Ph.D. Dissertation, Massachusetts Institute of Technology.
Taglicht, J. 1984. *Message and Emphasis.* Longman, New York.
von Stechow, A. 1984. Comparing Semantic Theories of Comparison. *Journal of Semantics,* 3: 1-77.

Multi-Site Data Collection and Evaluation in Spoken Language Understanding

*L. Hirschman, M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann**

Contact: Lynette Hirschman
NE43-643 Spoken Language Systems Group
MIT Laboratory for Computer Science, Cambridge, MA 02139
e-mail: lynette@goldilocks.lcs.mit.edu

ABSTRACT

The Air Travel Information System (ATIS) domain serves as the common task for DARPA spoken language system research and development. The approaches and results possible in this rapidly growing area are structured by available corpora, annotations of that data, and evaluation methods. Coordination of this crucial infrastructure is the charter of the Multi-Site ATIS Data COllection Working group (MADCOW). We focus here on: selection of training and test data, evaluation of language understanding, and the continuing search for evaluation methods that will correlate well with expected performance of the technology in applications.

1. Introduction

Data availability and evaluation procedures crucially structure research possibilities: the type and amount of training data affects the performance of existing algorithms and limits the development of new algorithms; and evaluation procedures document progress, and force research choices in a world of limited resources. The recent rapid progress in spoken language understanding owes much to our success in collecting and distributing a large corpus of speech, transcriptions and associated materials based on human-machine interactions in the air travel domain. MADCOW has coordinated the multi-site data collection and evaluation effort. The DARPA Spoken Language community has long recognized that we were simultaneously developing evaluation methodologies and relying on these methods to evaluate systems and to push the research forward. This tight feedback loop has permitted us to extend our evaluation methodology incrementally. This paper reports on the status of the MADCOW-coordinated data collection effort and on recent evaluations.

The multi-site data collection paradigm [3, 4] distributes the burden of data collection, provides data rapidly, educates multiple sites about data collection issues, and results in a more diverse pool of data than could be obtained with a single collection site. The resulting data represents a wide range of variability in speaker characteristics, speech style, language style and interaction style. It has allowed individual sites to experiment with data collection methods: replacing various system components with a human results in data we can aim for in the future, while completely automated systems help us to focus on the major current issues in system accuracy and speed. Sites have also experimented with interface strategies (spoken output only, tabular output only, response summaries, paraphrase, degree initiative taken by the system may be more or less appropriate for different users and different tasks and all can dramatically affect the type of data resulting).

MADCOW's recent accomplishments include:

- Release of 14,000 utterances for training and test, including speech and transcriptions;
- Release of almost 10,000 annotated utterances (7000 training utterances and three test sets of 2300 utterances total), balanced by site;
- A bug reporting and bug fix mechanism, to maintain the quality and consistency of the training data;
- An evaluation schedule that delivered training data and froze changes in the principles of interpretation[1] several months before the evaluation;
- An experiment with "end-to-end" evaluation that permits evaluation of system aspects not previously possible.

Table 1 shows the breakdown of all training data and Table 2 shows the breakdown for just the annotated data[2].

2. Current Evaluation Methodology

When the ATIS task was developed in 1990 [9], little work had been done on formal evaluation of under-

*This paper was written the auspices of the the Multi-Site ATIS Data Collection Working group (MADCOW). In addition to the authors, many other people, listed under the Acknowledgements section, made important contributions to this work.

[1] These are the principles that define how various vague or difficult phrases are to be interpreted; see section 2.1 below.

[2] A class A utterance can be interpreted by itself, with no additional context; a class D utterance requires an earlier "context-setting" utterance for its interpretation; and a class X utterance cannot be evaluated in terms of a reference database answer.

Site	Speakers	Scenarios	Utterances
AT&T	50	176	1887
BBN	62	307	2277
CMU	43	196	2480
MIT	75	250	2265
MIT: old DB	96	320	2940
SRI	781	130	3126
TOTAL	407	1379	13975

Table 1: Multi-site ATIS Data Summary

standing for natural language interfaces [3]. In the absence of a generally accepted semantic representation, the DARPA SLS community focussed instead on "the right answer," as defined in terms of a database query task (air travel planning). This permitted evaluation by comparing "canonical" database answers to the system answers using a comparator program [1]. There was consensus that coming to agreement on what constituted the right set of data from a database for any query answerable via database retrieval (given proper definitions of terms) would be far easier than coming to agreement on a standard semantic representation.

The original evaluation methodology was defined only for context-independent (*class A*) utterances. However, this left approximately half the data as unevaluable (see Table 2). Over the next two years, the evaluation method was extended to cover context-dependent queries (*class D* utterances), it was tightened by requiring that a correct answer lie within a minimal answer and a maximal answer (see section 2.1), and it was made more realistic by presenting utterances in scenario order, as spoken during the data collection phase, with no information about the class of an utterance. Thus, we now can evaluate on approximately 75% of the data (all non-*class X* data – see Tables 2 and 4). Because, at least in some applications, wrong answers may be worse than "no answer" we have used a *Weighted Error* metric follows:[4]

$$WeightedError = \#(No\_Answer) + 2 * \#(Wrong\_Answer).$$

2.1. The Evaluation Mechanism

The comparator-based evaluation method compares human annotator-generated canonical ("reference") database answers to system generated answers. The annotators first classify utterances into context-independent (A), context-dependent (D) and unevaluable (X) classes. Each evaluable utterance (class A or D) is then given minimal and maximal reference answers. The minimal reference answer is generated using NLParse[5] and the maximal answer is generated algorithmically from the minimal answer. A correct answer must include all of the tuples contained in the minimal answer and no more tuples than contained in the maximal answer.

The Principles of Interpretation provides an explicit interpretation for vague natural language expressions, e.g., "red-eye flight", "mid-afternoon," and specifies other factors necessary to define reference answers, e.g., how context can override ambiguity in certain cases, or how utterances should be classified if they depend on previous unevaluable utterances. This document is a point of common reference for the annotators and the system developers, and permits evaluation of sentences that otherwise would be too vague to have a well-defined database reference answer. The initial Principles of Interpretation was implemented in 1990. The document is now about 10 pages long, and includes interpretation decisions based on some 10,000 ATIS utterances. The document continues to grow, though over time fewer new issues arise. It is remarkable that such a small document has sufficed to provide well-defined interpretations for a corpus of this size. This demonstrates that rules for the interpretation of natural language utterances, at least in the ATIS domain, can be codified well enough to support an automatic evaluation process. Because this procedure was explicit and well-documented, two new sites were able to participate in the most recent evaluation (November 1992).

[3] This coincides with the beginnings of formal evaluation for written text, via the Message Understanding Conferences (MUCs) [8]. The MUC evaluation uses a domain-specific filled template as the basis for evaluation. To date, the goal of a domain-independent semantic representation, perhaps analogous to the minimal bracketing of the Penn Treebank database [2] for parsing, remains elusive.

Site	Class A	Class D	Class X	Total
ATT	396 37.4%	416 39.3%	247 23.3%	1059 14.8%
BBN	858 56.2%	357 23.4%	312 20.4%	1527 21.4%
CMU	539 37.5%	324 22.6%	573 39.9%	1436 20.1%
MIT	663 37.7%	680 38.7%	414 23.6%	1757 24.6%
SRI	607 44.8%	582 43.0%	166 12.3%	1355 19.0%
Total	3063 42.9%	2359 33.1%	1712 24.0%	7134 100.0%

Table 2: Distribution of the Annotated Training Data

[4] The decision to call a wrong answer twice as bad as not answering was made to reflect an intuition that misinformation was worse than explicit refusal to answer. However, a recent experiment [5] showed that for one system, subjects were able to detect a system error without losing any additional turns in 90% of the cases. In the remaining 10%, a system error caused the subject to lose several turns before recovering, leading to a reduced estimated weighting factor for system errors of 1.25.

[5] NLParse is a database access product of Texas Instruments.

2.2. Testing on the MADCOW Data

The test data selection procedure was designed to ensure a balanced test set. Test data for the November 1992 evaluation were chosen using procedures similar to those for the November 1991 test [3]. As sites submitted data to NIST, NIST set aside approximately 20% of the utterances to create a pool of potential test data; some 1200 utterances were included in the November 1991 test set; 1300 utterances were included in the November 1992 test set.

NIST's goal was to select approximately 1000 test utterances from the test data pool, evenly balanced among the five collection sites (AT&T, BBN, CMU, MIT, and SRI). Utterances were selected by session, i.e., utterances occurring in one problem-solving scenario were selected as a group, avoiding sessions that seemed to be extreme outliers (e.g., in number of class X utterances, total number of utterances, or number of repeated utterances). Because the test pool contained only marginally more utterances than were needed for the test, it was not possible to simultaneously balance the test set for number of speakers, gender, or subject-scenarios. The test set contained 1002 utterances. The breakdown of the data is shown in Table 3.

NIST verified and corrected the original transcriptions. However, some uncertainty about the transcriptions remained, due to inadequacies in the specifications for the transcription of difficult-to-understand speech, such as *sotto voce* speech. After the transcriptions were verified, the data were annotated by SRI to produce categorizations and reference answers. A period for adjudication followed the test, where testing sites could request changes to the test data categorizations, reference answers, and transcriptions. The final post-adjudication classification summary is shown in Table 4. Final evaluation results are reported in [6].

Collecting Site	Speakers	Scenarios	Utterances
ATT	7; 1M/ 6F	22	200
BBN	7; 3M/ 4F	28	201
CMU	4; 4M/ 0F	12	200
MIT	10; 3M/ 7F	37	201
SRI	9; 5M/ 4F	19	200
Total	37; 16M/21F	118	1002

Table 3: Multi-site ATIS Test Data November 1992

3. Limitations of the Current Evaluation

The current data collection and evaluation paradigm captures important dimensions of system behavior. However, we must constantly re-assess our evaluation procedures in terms of our goals, to insure that our evaluation procedures can help us assess the suitability of a particular technology for a particular application, and to insure that benchmark scores will correlate well with user satisfaction and efficiency when the technology is transferred to an application.

The advantage of using a pre-recorded corpus for evaluation is clear: the same data are used as input to all systems under evaluation, and each system's set of answers is used to automatically generate a benchmark score. This approach provides a uniform input across all systems and removes human involvement from the benchmark testing process (except that human annotators define the reference answers). Any annotated set of data can be used repeatedly for iterative training. However, some of these same strengths impose limitations on what we can evaluate.

First, there is the issue of the match between the reference answer and the user's need for useful information. The method can count answers as correct despite system misunderstanding: e.g., a system misrecognition of "Tuesday" that substitutes "Wednesday" may in a paraphrase of the understanding lead the user to believe the answer is wrong, but if all flights have daily departures, the database answer will be *canonically* correct. On the other hand, useful (but not strictly correct) answers will be counted wrong, because there is no "partially correct" category for answers.

Second, mixed initiative in human-machine dialogue will be required for technology transfer in many spoken language understanding applications. But the evaluation paradigm actively discourages experimentation with mixed initiative. A query that is a response to a system-initiated query is classified as unevaluable if the user's response can only be understood in the context of the system's query. During evaluation, any system response that is a query will automatically be counted as incorrect (since only database answers can be correct).

The use of pre-recorded data also preserves artifacts of the data collection system. For example, much of the test data were collected using systems or components of systems to generate responses that are presumed to be less accurate than a human would be. As a result, the data include many instances of system errors that affect the user's next query. A user may have to repeat a query several times, or the user may correct some error that the data collection system (but not the system under evaluation) made. These are artificial phenomena that would disappear if the data collection and evaluation systems were identical.

Site	Class A	Class D	Class X	Total
ATT	48 (24.0%)	41 (20.5%)	111 (55.5%)	200 (20.0%)
BBN	97 (48.3%)	27 (13.4%)	77 (38.3%)	201 (20.1%)
CMU	76 (38.0%)	66 (33.0%)	58 (29.0%)	200 (20.0%)
MIT	100 (49.8%)	67 (33.3%)	34 (16.9%)	201 (20.1%)
SRI	106 (53.0%)	46 (23.0%)	48 (24.0%)	200 (20.0%)
Total:	427 (42.6%)	247 (24.7%)	328 (32.7%)	1002 (100.0%)

Table 4: Breakdown of Test Data by Class

Finally, the current paradigm does not take into account the speed of the response, which greatly affects the overall interaction. Demonstration systems at several sites have begun to diverge from those used in benchmark evaluations, in part, because the requirements of demonstrating or using the system are quite different from the requirements for generating reference database answers.

These limitations of the comparator-based evaluation preclude the evaluation of strategies that are fundamental research issues and that are likely to be crucial in technology transfer. In particular, we need to develop metrics that keep human subjects in the loop and support human-machine interaction. However, the use of human subjects introduces new issues in experimental design. Over the past year, MADCOW has begun to address these issues by designing a trial *end-to-end* evaluation.

4. End-to-End Evaluation Experiment

The end-to-end evaluation, designed to complement the comparator-based evaluation, included 1) objective measures such as timing information, and time to task completion, 2) human-derived judgements on correctness of system answers and user solutions (*logfile evaluation*), and 3) a user satisfaction questionnaire.

The unit of analysis for the new evaluation was a scenario, as completed by a single subject, using a particular system. This kept the user in the loop, permitting each system to be evaluated on its own inputs and outputs. The use of human evaluators allowed for assessing partial correctness, and provided the opportunity to score other system actions, such as mixed initiatives, error responses and diagnostic messages. The end-to-end evaluation included both task-level metrics (whether scenarios had been solved correctly and the time it took a subject to solve a scenario) and utterance-level metrics (query characteristics, system response characteristics, the durations of individual transactions).

4.1. Experimental Design

An experimental evaluation took place in October 1992, to assess feasibility of the new evaluation method. We defined a common experimental design protocol and a common set of subject instructions (allowing some local variation). Each site submitted to NIST four travel planning scenarios that had a well-defined "solution set". From these, NIST assembled two sets of four scenarios. Each site then ran eight subjects, each doing four scenarios, in a counter-balanced design. Five systems participated: the BBN, CMU, MIT and SRI spoken language systems, and the Paramax system using typed input.

4.2. Logfile Evaluation

A novel feature of the end-to-end experiment was the *logfile evaluation*. This technique, developed at MIT [7], is based on the logfile which records and timestamps all user/system interactions. A human evaluator, using an interactive program,[6], can review each user/system interaction and evaluate it by type of user request, type of system response, and correctness or appropriateness of response. For user requests, the following responses were distinguished: 1) New Information, 2) Repeat, 3) Rephrase, or 4) Unevaluable. For system responses, the evaluators categorized each response as follows:

Answer: further evaluated as *Correct, Incorrect Partially Correct or Can't Decide;*
System Initiated Directive: further evaluated as *Appropriate, Inappropriate, or Can't Decide;*
Failure-to-Understand Message: no further evaluation;
Diagnostic Message: further evaluated as *Appropriate, Inappropriate, or Can't Decide.*

The evaluator also assessed the scenario solution, according to whether the subject finished and whether the answer belonged to the defined solution set.

To facilitate determination of the correctness of individual system responses, we agreed to follow the Princi-

[6]The program was developed by David Goodine at MIT; the evaluator instructions were written by Lynette Hirschman, with help from Lyn Bates, Christine Pao and the rest of MADCOW.

ples of Interpretation, at least to the extent that an answer judged correct by these Principles would not be counted incorrect. For this experiment, logfile evaluation was performed independently by Bill Fisher (NIST) and Kate Hunicke-Smith (SRI Annotation), as well as by volunteers at MIT and BBN. This gave us experience in looking at the variability among evaluators of different levels of experience. We found that any two evaluators agreed about 90% of the time, and agreement among multiple evaluators decreased proportionally.

5. Lessons Learned

The experiment provided useful feedback on the risks and advantages of end-to-end evaluation, and provides the basis for a refined evaluation procedure. For the initial trial, we made methodological compromises in several areas: a small number of subjects, no control over cross-site subject variability, few guidelines in developing or selecting scenarios. These compromises seemed reasonable to get the experiment started; however, the next iteration of end-to-end evaluation will need to introduce methodological changes to provide statistically valid data.

5.1. Sources of Variability

Valid comparisions of systems across sites require control over major sources of variability, so that the differences of interest can emerge. The use of human subjects in the evaluation creates a major source of variability, due to differences in the subjects pools available at various sites and the characteristics of individuals. We can minimize some of these differences by, for example, by training all subjects to the same criterion across sites (to account for differences in background and familiarity with the domain), by using many subjects from each site (so that any one subject's idiosyncrasies have less of an effect on the results), and by ensuring that procedures for subject recruitment and data collection across sites are as similar as possible (we made a serious effort in this direction, but more could be done to reduce the cross-site variability that is otherwise confounded with the system under evaluation). An alternative would be to perform the evaluation at a common site. This would allow for greater uniformity in the data collection procedure, it could increase the uniformity of the subject pool, and would allow use of powerful experimental techniques (such as within-subject designs). Such a common-site evaluation, however, would pose other challenges, including the port of each system to a common site and platform, and the complex design needed to assess potential scenario order effects, system order effects, and their interaction.

Another source of variability is the set of travel planning scenarios the subjects were asked to solve. Certain scenarios posed serious problems for all systems; a few scenarios posed particular problems for specific systems. However, the data suggest that there was a subset that could perform a reasonable diagnostic function.

5.2. Logfile Evaluation

Somewhat unexpectedly, we found that logfile evaluation was a useful tool for system developers in identifying dialogue-related problems in there systems. The evaluator interface allowed for rapid evaluation (about 5-15 minutes per scenario). However, the evaluator instructions appear to need refinement, the interface needs minor extensions, and most important, we need to design a procedure to produce a statistically reliable logfile evaluation score. In addition to the methods for achieving this reliability that have been outlined in the previous section, we would also like to consider combining assessments from evaluators.

A remaining thorny problem is the definition of *correct*, *partially correct*, and *incorrect answers*. For this experiment, we used the Principles of Interpretation document to define a correct answer, so that we would not need to develop a new document for these purposes. For the next evaluation, we need definitions that reflect utility to the user, not just "canonical" correctness.

Finally, we found that we could not rely on subjects to correctly complete the scenarios presented to them. In some cases, the subject was not able to find the answer, and in other cases, the subject did not follow directions regarding what information to provide in the answer. This made it difficult to compute accurate statistics for scenario-level metrics such as task completion and task completion time; this problem was exacerbated by the limited amount of data we collected.

5.3. Summary and Conclusions

Our goal in end-to-end evaluation is to create a procedure that accurately assesses the usability of current spoken language technology and provides useful feedback for the improvement of this technology. To be useful, the procedure must reliably identify differences between systems and must embody a clear understanding of which system attributes are desirable and should be improved over time. In developing evaluation procedures that involve human interactions, we need to carefully assess the validity of the measures we use. For example a measure such as the number of utterances per scenario may seem relevant (e.g., the subject was frustrated with answers and had to repeat a question several times), but in fact may reflect irrelevant aspects of the process (the subject was intrigued by the system and wanted to push its lim-

its in various ways). Meaningful evaluation will require metrics that have been systematically investigated and have been shown to measure relevant properties.

6. Future Directions

We view evaluation as iterative; at each evaluation, we assess our procedures and try to improve them. The comparator-based evaluation is now stable and the November 1992 evaluation ran very smoothly. We plan to continue our experiments with end-to-end evaluation, to work out some of the methodological problems described in the previous section. In addition, work on database expansion and portability will affect onging data collection and evaluation efforts.

The *ATIS relational database* has been expanded from 11 cities to 46 cities, to provide a more realistic task supporting more challenging scenarios. The new database was constructed using data from the Official Airline Guide and now includes 23,457 flights (compared to 765 flights). The set of new cities was limited to 46 because it was felt that a larger set would result in an unwieldy database and would thus require the sites to devote too many resources to issues peripheral to their research, such as database management and query optimization. Data collection on this larger database is now beginning.

The *portability* of the technology (from application to application, and from language to language) becomes an increasing challenge as the technology improves, since more potential applications become possible. It still takes many hours of data collection and several person months of system development to port an application from one domain (e.g., air travel) to another similar domain (e.g., schedule management). Evaluating portability is still more challenging. Evaluation has a significant cost: the comparator-based method requires the definition of a training corpus and its collection, defining principles of interpretation, and (most expensively) the annotation of data. Therefore, if we believe that regular evaluations play an important role in guiding research, we need to find cost-effective ways of evaluating systems. End-to-end evaluation can provide some low-overhead techniques for quickly evaluating system performance in new domains.

6.1. Conclusion

MADCOW has played a central role in developing and coordinating the multi-site data collection and evaluation paradigm. It will also play an active role in defining new methodologies, such as end-to-end evaluation, to support evaluation of interactive spoken language systems. We believe that end-to-end evaluation will allow us to assess the trade-offs among various component-level decisions (in speech recognition, natural language processing and interface design), bringing spoken language systems closer to eventual deployment.

7. ACKNOWLEDGEMENTS

We would particularly like to acknowledge the contribution of Nancy Dahlgren at NIST; prior to her accident, Nancy made important contributions to the annotation and debugging of the data. We greatly missed her participation during the final evaluation.

In addition to the authors, the following people made a valuable contribution to the process: at BBN: R. Bobrow, R. Ingria, J. Makhoul, V. Shaked, and D. Stallard; at CMU: C. Neelan, E. Thayer, and R. Weide; at MIT: D. Goodine, J. Polifroni, C. Pao, M. Phillips, and S. Seneff; at NIST: N. Dahlgren, J. Fiscus, and B. Tjaden; at Paramax: L. Norton, and R. Nilson; and at SRI: H. Bratt, R. Moore, E. Shriberg, and E. Wade.

References

1. Bates, M., S. Boisen, and J. Makhoul, "Developing an Evaluation Methodology for Spoken Language Systems," *Proc. Third DARPA Speech and Language Workshop*, R. Stern (ed.), Morgan Kaufmann, June 1990.
2. Black, E., *et al.*, "A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars," *Proc. Third DARPA Speech and Language Workshop*, P. Price (ed.), Morgan Kaufmann, June 1991.
3. Hirschman, L., *et al.*, "Multi-Site Data Collection for a Spoken Language Corpus", *Proc. Fifth Speech and Natural Language Workshop*, ed. M. Marcus, Morgan Kaufmann, Arden House, NY, February 1992.
4. L. Hirschman *et al.*, "Multi-Site Data Collection for a Spoken Language Corpus," *Proc. of the ICSLP*, Banff, Canada, October 1992.
5. L. Hirschman and C. Pao, "The Cost of Errors in a Spoken Language Systems," submitted to Eurospeech-93, Berlin 1993.
6. Pallett, D., Fiscus, J., Fisher, W., and J. Garofolo, "Benchmark Tests for the Spoken Language Program," *Proc. DARPA Human Language Technology Workshop*, Princeton, March 1993.
7. Polifroni, J., Hirschman, L., Seneff, S. and V. Zue, "Experiments in Evaluating Interactive Spoken Language Systems" *Proc. DARPA Speech and Natural Language Workshop*, Arden House, NY, February 1992.
8. *Proc. Fourth Message Understanding Conf.*, Morgan Kaufmann, McLean, June 1992.
9. Price P., "Evaluation of Spoken Language Systems: The ATIS Domain," *Proc. Third DARPA Speech and Language Workshop*, P. Price (ed.), Morgan Kaufmann, June 1990.

INTEGRATING TWO COMPLEMENTARY APPROACHES TO SPOKEN LANGUAGE UNDERSTANDING

Eric Jackson

SRI International
333 Ravenswood Ave., Menlo Park, CA 94025

ABSTRACT

A current goal in spoken language understanding research is to combine the robustness of domain-specific template fillers (e.g., script and case frame-based systems) with the syntactic coverage of parser-based systems. This paper describes an integration of a pair of systems representing each of these types into a new system that takes advantage of their complementary strengths.

INTRODUCTION

Building a natural language system for written text with high coverage is notoriously difficult; the range and variability of human language are tremendous. The task of building a *spoken* language system, however, poses an additional set of challenges. For one thing, current speech recognition systems are far from perfect; a 10% word-recognition error rate over a 1000-word domain is considered very good. Furthermore, difficult phenomena such as run-on sentences, fragments, false starts, flexible constituent ordering and infelicitous word choice are especially prevalent in spontaneous spoken language. These phenomena pose difficult problems for parser-based natural language understanding systems, because the grammars they use are often designed to expect well-formed complete sentences.

Template filling systems typically fill slots in domain-specific templates by matching fixed patterns against portions of an input string. These systems may be able to produce interpretations without accounting for every word in the utterance, and without knowledge of the syntactic structure of the entire utterance. This fact lends these systems a degree of robustness to recognition errors, and to the difficult and unexpected phenomena often encountered in spontaneous speech.

It is often impossible, however, to correctly interpret a sentence without knowing its syntactic structure. The syntactic relation of a word or phrase to other parts of the sentence may be impossible to determine from the local context of the word or phrase, and yet may be necessary for grasping the meaning of the sentence. The work reported here attempts to combine into a single system the capabilities of parser-based natural language systems with the robustness of template fillers. Similarly motivated work has been done on building semantic interpretations from partial parses (e.g., [1, 2]). Our work differs primarily in that our basic interpretation mechanism is template filling.

The systems described here have been developed for the Air Travel Information System (ATIS) task. This is the common task for sites participating in the DARPA Spoken Language Systems project. The systems have been developed and tested on actual spontaneous speech data collected from naive users presented with air travel planning scenarios. The ATIS corpus currently consists of over 10,000 such spoken utterances.

THE TEMPLATE MATCHER

The Template Matcher [3] was developed at SRI specifically to handle the sorts of spontaneous speech phenomena that are difficult for a parser-based system. The main operation of the Template Matcher is the filling in of domain-specific templates. For example, templates for the air travel domain include the flight, fare and ground transportation templates. Templates are associated with slots, which are filled with information from the input sentence. So, for example, the input sentence "Show me the flights from Boston to Dallas on United" produces the following template:

[flight, [origin, BOSTON], [destination, DALLAS], [airline, UNITED]]

Slots are filled by matching fixed phrases against the input sentence. The origin slot, for example, may be filled if part of the sentence matches any of the following patterns: "from <airport-or-city>," "out of <airport-or-city>" or "between <airport-or-city> and <airport-or-city>."

The system builds a template with the set of slots that maximizes use of the words of the sentence (ignoring small function words). It then assigns a score to the resulting filled template, which reflects how much of the input sentence was used in building the template. The higher the score, the greater the likelihood that the template is correct. The system decides whether to answer the query by comparing the score to a threshold parameter. This threshold allows for a certain amount of risk trade-off. The system can set the threshold low to maximize the number of correct answers, or it can set the threshold high to minimize the number of wrong answers.

The ability of the Template Matcher to successfully interpret most spontaneous speech in the ATIS domain is testified to by the results of the latest DARPA benchmark tests. On the natural-language-only test, where

systems are given the correct transcription of each utterance as input, the system achieved the following results:

Right	Wrong	No Answer
533	60	94

while on the spoken language system test, where utterances are passed through the speech recognizer to the natural language component, the results were:

Right	Wrong	No Answer
444	69	174

(Systems are evaluated on a subset of the ATIS corpus that has not previously been seen by system developers. Responses are evaluated by comparing them with the database answers produced by trained annotators. A wrong answer is considered twice as bad as no answer.)

Despite these promising results, it is clear that the Template Matcher is incapable of correctly interpreting many types of utterances, since it has no knowledge of syntactic structure. Although it might be possible to extend the Template Matcher to have extremely high coverage of actual speakers' utterances in the ATIS domain, its coverage could never be perfect. Furthermore, in other domains it might not be possible to obtain as good coverage with the same sort of system.

GEMINI

In parallel with the development of the Template Matcher, SRI has been developing a parser-based natural language system known as "Gemini." Thus, an obvious approach to building a system that combined robust interpretation with parsing capabilities was to integrate these two existing systems whose strengths are complementary.

Gemini, an extension and reimplementation of the Core Language Engine [4], is based upon an efficient bottom-up parser and a domain independent unification grammar. It incorporates a bottom-up parser [5] so that an integrated system of the sort we are describing here can be successful when the parser is unable to parse the entire input utterance. For a pure bottom-up parser finds all the structure it can in the utterance, while a top-down, left-to-right parser ceases to find structure beyond the point in the sentence where it gets stuck. In many cases, the partial structure that only a bottom-up parser would find is what is needed to help the Template Matcher correctly interpret the utterance.

Unfortunately, pure bottom-up parsing is inefficient, largely because of the problem of gaps. Gaps are positions in an input sentence where a category is realized by the empty string. For example, in the sentence "What cities does American fly to from Boston?" an empty noun phrase appears between "to" and "from." Gaps must be filled by material elsewhere in the sentence; in our example, the phrase "what cities" fills the NP gap. In contrast, the string "American flies to" will not parse because, although an NP gap may be hypothesized after the word "to," there is nothing to fill it.

Since a bottom-up parser does not impose any left-context constraints, it will hypothesize every possible gap at every position of a sentence, even when there are no potential gap-fillers. Since gaps do not require a match against the input string, this results in a large number of hypotheses that lead nowhere, which, in turn, negatively impacts parsing efficiency. Gemini avoids this problem by not doing *pure* bottom-up parsing. It imposes limited top-down constraints to cut down on the proliferation of hypotheses. The set of categories in the grammar are partitioned into those that are context-dependent and those that are context-independent. Context-dependent categories are hypothesized only when predicted by left context. For example, categories containing wh-gaps (such as a verb phrase with an object gap) are context dependent; they are only hypothesized if a wh-element appears earlier in the sentence. The Gemini parser can easily be parameterized to impose more or fewer top down constraints.

Gemini also can apply sorts restrictions as it parses. So, for example, although the sentence, "How many flights fly on large aircraft after five PM?" has at least two possible parses (depending on whether "after five PM" modifies "aircraft" or "fly"), Gemini knows that a time restriction may modify a flying event, but not an aircraft, so if sorts restrictions are applied, only one of the two parses will be produced.

On a corpus of 2139 sentences, Gemini generates at least one parse for 2039 (95%). It generates parses that meet sorts restrictions for 1915 (90%).

INTEGRATION OF THE TWO SYSTEMS

The new system is an enhancement of the Template Matcher that makes use of structural information found by Gemini in building templates. Filled slots are identified by pattern matches against the input string, and then passed up the phrase structure tree during which they may be combined or altered if certain conditions obtain. Two mechanisms are invoked to select the best parse when there are multiple candidates: 1) the sorts package mentioned above which filters out many semantically anomalous parses, and 2) a syntactic parse-preference mechanism that implements an algorithm due to Pereira favoring low attachment [6]. A template is then built that accounts for as many words and constructions in the input as possible. This approach allows us to handle problems such as modifier attachment and scope resolution that would be difficult or impossible for a pure template-matching system.

For example, consider the problem of modifier attachment posed by the following sentence:

"Show me flights arriving in Boston on 747s before ten."

The problem for the Template Matcher is that the time specification "before ten" should constrain the flight arrival time, but there is no way to tell this by looking at the phrase "before ten" in isolation. To know that the restriction constrains the arrival time, and not the departure time, the system needs to know that "before

ten" modifies "arriving."

In order to see how the system works, let us examine how the system would process the above example.

Step 1: The Template Matcher locates all the phrases that match slot-filling patterns (see Figure 1). Notice that a phrase may fill more than one slot. For example, the phrase "in Boston" may fill the ground_city slot of a ground transportation template (e.g., "Show me ground transportation in Boston"), or the "in" slot of a flight template. The "arrive" and "in" slots are actually special temporary slots; they will never appear in a template. In certain circumstances, as we shall see, these slots may combine to form a destination slot, which does appear in the final template.

Step 2: The Gemini parser is run and builds the left two parse trees shown in Figure 2. The tree on the far right will not be built because it violates sorts restrictions (time restrictions do not modify 747s).

Step 3: When we pass slots up the two parse trees, the "arrive" and "in" slots combine to form a "destination" slot with "BOSTON" as its filler. In addition, in one tree, the "arrive" and "before" slots combine to form an "arriving_before" slot, while in the other tree, the "before" slot turns into a "departing_before" slot, because that is the default. The nodes where slots are combined or altered are circled (Figure 2).

Step 4: Since a syntactic ambiguity has produced conflicting sets of slots, the parse preference mechanism is now invoked, which favors "low attachment." (It also favors something called "minimal attachment" which will not be discussed.) We will render the notion of low attachment precise below, but the basic idea should be clear; the tree in the middle in Fig. 2 is preferred because the "before ten" phrase is attached lower than in the tree on the left.

The main insight of the Pereira paper [6] on which the parse preference mechanism is based was that low attachment could be optimized in a shift-reduce parser by preferring shifts over reduces during the course of parsing. This preference, however, can be equally well be imposed as a *post hoc* comparison metric on complete parse trees. To compare two parse trees, find the sequences of shift-reduce operations needed to produce those trees, then find the first position at which there is a shift operation in one sequence and a reduce operation in the other. The tree corresponding to the sequence with the shift operation is the preferred one.

To construct the sequence of shift and reduce operations corresponding to a given parse tree, do a post-order traversal of the tree, and, at each step, add a shift to the sequence, if the current node is a leaf, and, add a reduce, if it is not. Applying this procedure to the two parse trees above (and ignoring unit reductions), we see

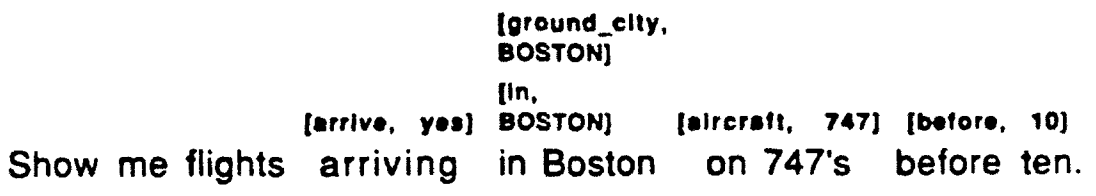
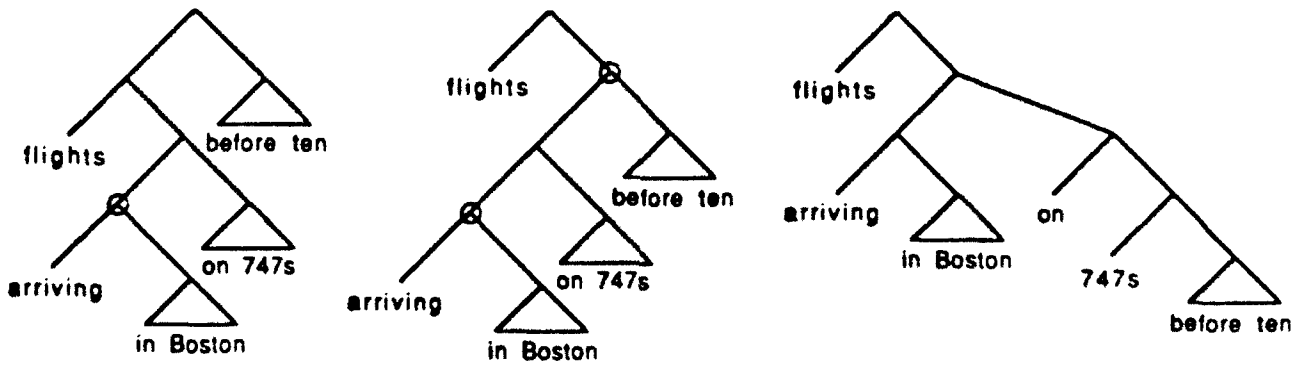


Fig. 1 - Filled slots for the example.



[[destination, BOSTON],
[aircraft, 747],
[departing_before, 10]]

[[destination, BOSTON],
[aircraft, 747],
[arriving_before, 10]]

Figure 2 - Parse trees for the example.

that the tree on the left corresponds to the sequence "SSSSRRSSRRSSRR," while the tree in the middle corresponds to the sequence "SSSSRRSSRRSSRRR." Since, in the eleventh position, the first sequence has a reduce while the second sequence has a shift, the second sequence (and hence the tree in the middle) is preferred.

Step 5: Now that the appropriate set of slots has been found, the system inserts them in a flight template and passes that template to the database query generation component. A database query is produced and the appropriate data are retrieved and presented to the user.

RESULTS

From a corpus of spoken utterances collected from naive users in the air travel domain, we assembled a test set of 64 utterances that raised nontrivial modifier attachment problems. Of these 64 sentences, the system we describe here was able to correctly interpret 57. The remaining 7 sentences do not raise great difficulties; most could be handled with simple improvements to the grammar.

OTHER APPLICATIONS

Preliminary work has been done to handle problems of scope resolution. To allow templates to reflect scope information, the syntax of templates was extended. For example, to indicate the scope of the negation in the sentence "I want American flights not leaving after five," the template produced is:

[flight, [airline, AMERICAN], [not, [[departing_after, [500, 1700]]]]]

An arbitrary number of slots can be embedded within the "not" operator. Similar constructs allow templates to reflect the scope of superlatives and coordinated constituents.

Without help from a parser, the Template Matcher would misinterpret many queries involving negation, since the word signalling the negation (i.e., "not") may be far removed from the negated constraint. A system that interprets phrases with regard to their immediate context only will be unable to handle negation in a general way. The integrated system handles negation scope in much the same way as it handles modifier attachment. Filled slots are passed up the phrase structure tree, and under the right conditions wrapped in the negation operator. For example, when a verb phrase is combined with the lexical item "not" to form a new verb phrase, the filled slots associated with the embedded verb phrase are embedded in the negation operator.

Of course, there are many cases where something more sophisticated is needed. Consider, for example, the query "Show me flights not arriving in Dallas after five P M." Presumably the correct response in this case is to show all flights that *do* have a destination of Dallas but do not arrive after five P M. Our current system would show, in addition, all the flights in the database not flying to Dallas. Despite this, the current treatment of negation seems to be an improvement over what was available with the Template Matcher alone.

ACKNOWLEDGMENTS

Gemini has been designed and built by Doug Appelt, John Bear, Lynn Cherny, John Dowding, Mark Gawron, Bob Moore and Doug Moran. Doug Appelt is responsible for the template-to-database-query translation code. Bob Moore implemented the parse preference mechanism.

This research was supported by the Defense Advance Research Projects Agency under Contract N00014-90-C-0085 with the Office of Naval Research.

REFERENCES

[1] S. Seneff. "A Relaxation Method for Understanding Spontaneous Speech Utterances", *Proceedings Fifth DARPA Workshop on Speech and Natural Language*, February 1992 (forthcoming).

[2] R. Weischedel. "Partial Parsing: A Report on Work in Progress", *Proceedings Fourth DARPA Workshop on Speech and Natural Language*, pp. 204-209, February 1991.

[3] E. Jackson, D. Appelt, J. Bear, R. C. Moore and A. Podlozny. "A Template Matcher for Robust NL Interpretation," *Proceedings Fourth DARPA Workshop on Speech and Natural Language*, pp. 190-194, February 1991.

[4] H. Alshawi (ed.). *The Core Language Engine*. Cambridge: The MIT Press, 1992.

[5] R. C. Moore and J. Dowding. "Efficient Bottom-Up Parsing," *Proceedings Fourth DARPA Workshop on Speech and Natural Language*, pp. 200-203, February 1991.

[6] F.C.N. Pereira. "A New Characterization of Attachment Preferences," in *Natural Language Processing: Psycholinguistic, Computational and Theoretical*. Cambridge: Cambridge University Press, 1983.

A Template Matcher for Robust NL Interpretation

Eric Jackson, Douglas Appelt, John Bear, Robert Moore, and Ann Podlozny

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Abstract

In this paper, we describe the Template Matcher, a system built at SRI to provide robust natural-language interpretation in the Air Travel Information System (ATIS) domain. The system appears to be robust to both speech recognition errors and unanticipated or difficult locutions used by speakers. We explain the motivation for the Template Matcher, describe in general terms how it works in comparison with similar systems, and examine its performance. We discuss some limitations of this approach, and sketch a plan for integrating the Template Matcher with an analytic parser, which we believe will combine the advantages of both.

Introduction

One of the conclusions SRI has drawn from working with the ATIS common task data is that, even with a very constrained user task, there will always be unanticipated expressions and difficult constructions in the spoken language elicted by the task that will cause problems for a conventional, analytical approach to natural-language processing. However, it also seems that requests for only a few types of information account for a very large proportion of the utterances produced by users performing a task like air travel planning. This point is illustrated by some of the more difficult queries in the June 1990 test set:

> Give me a list of all airfares for round-trip tickets from Dallas to Boston flying on American Airlines.
>
> Show me all the flights and their fares from San Francisco to Boston on June second.
>
> I need information on airlines servicing Boston flying from Dallas.

In the first example the phrase "flying on American Airlines" apparently modifies "tickets," with the flights that the tickets are for apparently being the implied subject of "flying." The second example seems to contain a discontinuous constituent, "flights ... from San Francisco to Boston on June second," which is the antecedent of the pronoun "their" that occurs in the middle of the discontinuity. The third example would be straightforward, except for the fact that the verb "servicing" has been substituted for the more conventional "serving." Despite the difficult linguistic problems posed by these queries, the information they request is very simple—just fares, flights, and airlines for travel between a pair of specified cities.

Consideration of examples such as these has led us to modify our approach to natural-language processing in spoken language systems. The key modification to our system is the addition of a Template Matcher to provide robust interpretation for the most common types of requests in the task domain. The Template Matcher achieves robustness in two ways: (1) it provides an interpretation when not all the words or constructions in an utterance have been accounted for, and (2) it provides a mechanism for trading-off the risk of wrong answers with the degree of coverage. These properties arise from a mechanism that assigns scores to interpretations, penalizing interpretations that do not account for words in the utterance. The bulk of this paper is devoted to describing the Template Matcher and discussing its performance as a stand-alone system for interpretation of natural-language queries for the ATIS task. Later in the paper we consider how such a module might best fit into a complete system for spoken-language understanding.

Description of the System

The Template Matcher operates by trying to build "templates" from information it finds in the sentence. Based on an analysis of the types of sentences observed in the ATIS corpus, we devised four templates that account for most of the data: flight, fare, ground transportation, and meanings of codes and headings. We have recently added several new templates, including aircraft, city, airline, and airport. Templates consist of slots which the Template Matcher fills with information contained in the user input. Slots are filled by looking through the sentence for particular kinds of short phrases. For example, "from" followed by an airport or city name will cause the "origin" slot to be filled with the appropriate name. The sentence

Show me all the United flights Boston to Dallas nonstop on the third of November leaving after four in the afternoon.

would generate the following flight template:

```
[flight,[stops,nonstop],
        [airline,UA],
        [origin,BOSTON],
        [destination,DALLAS],
        [departing_after,[1600]],
        [date,[november,3,current_year]]]
```

The template score is basically the percentage of words in the sentence that contribute in some way to the building of that template. Given an input sentence, the Template Matcher constructs one template of each sort, and the one with the best score is used to construct the database query, provided its score is greater than a certain "cut-off" parameter. The cut-off parameter is what permits the risk trade-off mentioned above: the higher the cut-off, the more conservative the system is in attempting to produce a response. Words can contribute to a score in different ways: words that fill a slot (e.g., "Boston") add to the score, words that help get a slot filled (e.g. "from") also add to the score. Some words may not contribute to the interpretation, but nonetheless confirm the choice of a particular template (e.g., "downtown" for the ground transportation template), and hence are added to the score for that template. Other words are ignored for the purposes of scoring (e.g., "and," "please," "ok," and "show"), since they do not tend to confirm particular templates.

In certain cases the Template Matcher may modify the basic score of a template. Each template has a set of key words (or key phrases). The presence of these words or phrases in a sentence is a strong indication that the associated template is the appropriate one for that sentence. For the flight template, the keywords include words like "flight," "fly," and "go"; for the fare template, words and phrases such as "how much," "fare," and "price" are examples; for the meaning template, examples include "what is," "explain," and "define." If none of a template's key words are present in a sentence then that template's score is docked by a certain keyword punishment factor, which varies from template to template. In most cases the lack of a keyword will prevent the associated template from scoring above the cut-off.

There are two situations in which the Template Matcher will "abort" a given template, that is, give it a score of zero and cease processing it. First, if the system tries to fill a slot in a certain template with two different values, that template is aborted. Since we have no better than a fifty-fifty chance of guessing which is the correct filler, we are better off not attempting any answer. Second, if a template has no slots filled, it will receive a score of zero. This restriction is relaxed when the Template Matcher is operating in "context-dependent" mode, where follow-up questions are expected. A query like "show me the fares," which would not fill any slots, would be much more likely as a follow-up question than as a context-independent query.

Comparison with Other Systems

Systems using the basic idea behind the Template Matcher go back as least as far as the SAM system at Yale [2], and include the Phoenix system at CMU [3, 4] and the SCISOR system at General Electric [5] as recent examples. There is also a degree of similarity to "case-frame"-based parsing methods [6, 7]. The main distinction is that the slots in our templates are domain-specific concepts rather than general linguistic or conceptual cases.

Of these precursors, the Phoenix system seems most similar to the Template Matcher. Like the Template Matcher, the Phoenix system has templates (which they call "frames") with slots that get filled with information from the sentence. The scoring mechanisms of the two systems are similar, but not identical. For both, the basic score of an interpretation is the number of words in the sentence that the interpretation accounts for. In the Phoenix system, for a word in a sentence to count for an interpretation's score, it must help fill some slot in that interpretation's frame. For the Template Matcher, the word will also count if it is an "ignore" or "confirm" word as discussed above.

There are several other differences between the scoring mechanisms of the two systems: The Template Matcher punishes templates that do not have a keyword present in the sentence, and the Template Matcher requires that at least one slot in a template be filled. Also, the two systems behave differently when an attempt is made to fill a single slot with two different fillers. The Template Matcher will abort a template if this happens, while the Phoenix system will fill the slot with the second of the two possible fillers. The latter approach will handle certain types of false starts, but might be expected to yield more incorrect answers in other situations. Finally, CMU is not currently using a cutoff to weed out bad interpretations, although given the existence of a scoring mechanism in their system, this is something they clearly could do.

Results

After two weeks of development this system was tested on the June 1990 ATIS test set. This was a fair test to the extent that the implementor of the matching routines and the templates themselves (Jackson) had not examined the data from this test set prior to the evaluation. (Moore had noted, however, that the test set queries seemed amenable to a template-matching approach). For various values of the cut-off parameter we obtained the results shown in the following table.

Cut-off	Right	Wrong	No Answer
0.000	55	13	22
0.833	42	4	44
1.000	37	2	51

(These results were determined by visual inspection of the templates; the database retrieval code was not implemented at this point.) The conclusion we drew from this test is that a template-matching approach could quickly yield results that were competitive with the some of the better results reported in the original June 1990 ATIS test.

After completing the implementation of the system and extensive development using the ATIS training data, we used the Template Matcher for the February 1991 ATIS class A evaluation, in both the NL and SLS tests. The results as measured by NIST are shown below.

Test	Right	Wrong	No Answer
NL only	109	9	27
SLS	96	11	38

We used a cut-off of 0.8 for this evaluation, as we had previously determined from training data that this value should come close to optimizing the number of right answers minus the number of wrong answers.

The system for the SLS tests was a serial connection of the version of SRI's DECIPHER system used in the ATIS SPREC evaluation and the Template Matcher described above. The answers reported in the SPREC evaluation were edited to be in lexical SNOR format and run through the Template Matcher exactly as in the NL tests. It is interesting to note the relatively small degradation from the NL to the SLS results, despite a 18.0 percent word error rate in the speech recognition; this seems to indicate the robustness of the Template Matcher to recognition errors.

We had not planned to participate in the D1 evaluation, but at the request of NIST, we did those tests as well, taking context into account by using the answer to the first query in the D1 pair to restrict the database search in answering the second query, the same technique used in our ATIS demo system. In addition, the Template Matcher was run in context-dependent mode for the second query of each D1 pair. The results on the second queries of the pairs as measured by NIST are shown in the table below.

Test	Right	Wrong	No Answer
NL only	22	3	13
SLS	15	11	12

We have not yet analyzed why there was a greater degradation in going from the NL to the SLS results in the D1 tests.

Limitations

In this section, we discuss some sentences that cause problems for the Template Matcher that are not easily resolvable.

> Show me flights returning from Dallas into San Francisco by ten P M.

This sentence is a good example of the need for syntactic information. The problem is that the Template Matcher cannot tell that the phrase "by ten P M" modifies "returning," and thus constrains the arrival time. By default, it treats the "by" phrase as restricting the departure time, and thus misinterprets the query.

> What is an A fare?

The problem here is that "A" is ambiguous; it may be either the indefinite article or a fare class code. We have been forced to leave the fare class code "A" out of the Template Matcher lexicon. Adding it would do more harm than good, for we would then misinterpret every occurence of the phrase "a fare" (with the indefinite article), as in "Give me a fare from Boston to Dallas." Syntactic information could help resolve this ambiguity, as could speech information, since the determiner "a" and the letter "A" have different acoustic properties.

> List the fares for Delta flight eight oh seven and Delta flight six twenty one from Dallas to Denver.

Conjunctions of complex noun phrases are beyond the scope of the Template Matcher as it currently stands. The system could be modified to handle such phenomena, but an analytical grammar might be the more natural tool for the job.

> Do you have to take a Y N flight only at night?

This is an example of a sentence where all the words contribute to a certain template (the flight template, in this case) and yet that template is not the correct one.

A New Architecture

As the examples in the previous section suggest, the Template Matcher by itself is probably not the ultimate solution to the problem of robust interpretation of natural-language queries. We believe that the template-matching approach and an analytical parser-based approach have complementary strengths and that an approach that combines both of them is likely to be ultimately superior than either one alone. We have therefore begun developing a new architecture for language processing in spoken language systems that combines the two approaches. Our basic strategy will be to use the analysis produced by the parser whenever we can, but to fall back on the Template Matcher when the parser-based system fails to produce a complete analysis. It is our conjecture, supported at least in part by the best results reported in the June 1990 ATIS evaluation, that an analytical, parser-based approach can be designed so that when it succeeds in providing a complete analysis of the input, that analysis has a very high probability

of being correct. With the Template Matcher it seems that there will inevitably be a larger possibility for error, because it uses strictly less of the information available in the utterance than a parser. In particular, our Template Matcher can ignore words; it ignores order; and it has almost no notion of structure. By using the Template Matcher as a backup to the parser-based system, we eliminate the possibility of the Template Matcher getting a wrong interpretation of something that could be successfully analyzed by the parser.

A second reason for running the Template Matcher after the parser is to enable the Template Matcher to use partial results of parsing in its operation. Our current Template Matcher uses only single words and fixed phrases as key words or slot fillers. We are in the process of extending the Template Matcher so that it uses whole phrases that have been identified by the parser in attempting to analyze the entire utterance. For example, we saw that the Template Matcher is unable to analyze a phrase as complex as "returning from Dallas into San Francisco by ten P M." Generalized to work from parsed phrases, the Template Matcher might be able to successfully interpret a complex utterance containing this phrase even if the entire utterance could not be parsed. Additionally, running the Template Matcher on parsed phrases should cut down on the sheer number of particular word patterns that have to be included in the template specifications.

The use of robust interpretation methods changes the way in which the constraints embodied in a grammar are viewed. They must be treated as soft, rather than hard, constraints. This has significant implications for the rest of a spoken language system. If we want the parser to find grammatical fragments of the input that may be of use to the Template Matcher, then the parsing algorithm we previously used, which imposed strong left-context constraints, is no longer appropriate. We want something closer to pure bottom-up parsing to find all the phrases that the Template Matcher might use. We have developed such a parser, whose details are outlined in another paper for this workshop [1].

Perhaps the most significant consequence of using robust interpretation methods in a spoken language system, however, is that the failure to find a complete parse can no longer be used as a hard constraint to reduce perplexity for the speech recognizer. An analytical grammar still contains valuable information that should be used by the recognizer, however. We feel that one promising approach to making use of this information is to extend the idea of a word-based statistical language model, such as a bi-gram model, to a phrase-based statistical language model, e.g., a "bi-phrase" model. The idea is simply to estimate the probability of occurrence of a particular type of phrase conditioned on the type of phrase that precedes it. In making this work effectively, however, it is important to include some lexical information in the categorization of phrases, usually information about the lexical head of the phrase.

The ability of such a framework to capture long distance constraints not captured by N-gram models is illustrated by an utterance such as "What airlines that serve Boston fly 747s?" If we want to predict the likelihood of "fly" occuring in this context, the preceding word "Boston" gives us essentially no information. If, however, we have identified "What airlines that serve Boston" as a noun phrase whose lexical head is "airlines" then the likelihood of a verb whose lexical head is "fly" should be relatively high.

The incorporation of a probabilistic element into the system raises a number of other interesting possibilities, including incorporation of probabilistic scoring based on observations of likelihoods of particular templates for sentences in the corpus, of particular slots for each template, and of particular words for each slot; and the possibility of using the Template Matcher itself as the basis of a statistical language model to guide recognition.

Summary

In sum, the Template Matcher represents a complementary approach to traditional natural-language processing. It has the virtues of robustness and broad coverage of many linguistic variants for requests for specific types of information. Although we have not discussed the issue of computational efficiency in this paper, the Template Matcher is noticably faster than a typical parser. The approach also has the advantage of rapid development time which should enhance portability to new domains.

Acknowledgments

This research was supported by the Defense Advanced Research Projects Agency under Contract N00014-90-C-0085 with the Office of Naval Research.

References

[1] Moore, R.C., and Dowding, J., *Efficient Bottom-Up Parsing*, Proceedings, Fourth DARPA Workshop on Speech and Natural Language, February 1991.

[2] Schank, R.C. and Yale A.I. Project, *SAM—A Story Understander*, Research Report 43, Department of Computer Science, Yale University, 1975.

[3] Ward, W., *Understanding Spontaneous Speech*, Proceedings, DARPA Speech and Natural Language Workshop, February 1989.

[4] Ward, W., *The CMU Air Travel Information Service: Understanding Spontaneous Speech*, Proceedings, DARPA Speech and Natural Language Workshop, June 1990.

[5] Rau, L.F., and Jacobs, P.S., *Integrating Top-Down and Bottom-Up Strategies in a Text Processing System*, Proceedings, Second Conference on Applied Natural Language Processing, Austin, Texas, 1988.

[6] Riesbeck, C., and Schank, R.C., *Comprehension by Computer: Expectation-Based Analysis of Sentences in Context*, Research Report 78, Department of Computer Science, Yale University, 1976.

[7] Carbonell, J.G. and Hayes, P.J., *Recovery Strategies for Parsing Extragrammatical Language*, Technical Report CMU-CS-84-107, Carnegie-Mellon University Computer Science Technical Report, 1984.

SRI's Experience with the ATIS Evaluation

Robert Moore, Douglas Appelt, John Bear, Mary Dalrymple, and Douglas Moran

SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025

Abstract

SRI International participated in the June 1990 Air Travel Information System (ATIS) natural-language evaluation. This report briefly describes the system that SRI used in the evaluation, analyzes SRI's results, and makes some recommendations for changes in the database structure and data collection system to be used for future ATIS evaluations.

The SRI ATIS System

The natural-language processing system used by SRI in the June 1990 ATIS evaluation is a derivative of the Core Language Engine (CLE) developed at SRI's Cambridge Research Centre in Cambridge, England [1]. At present, the main processing components of SRI's ATIS system are taken from the CLE, while the grammar, semantic interpretation rules, and lexicon are substantially new. The system divides query processing into the following phases:

- Lexical lookup
- Syntactic parsing
- Semantic interpretation and selectional filtering
- Quantifier scoping
- Database query generation
- Query optimization
- Database retrieval

The syntactic and semantic rules used in the parsing and interpretation phases are expressed in a unification-based formalism. The parser is based on a left-corner parsing algorithm for context-free grammar that has been generalized to apply to unification grammar by substituting unification for identity checks in dealing with gramatical category expressions. An attribute/value notation for feature constraints is provided for the grammar writer, but this notation is compiled into ordinary term structures by assigning, for each major category symbol, an argument position for each feature that can occur with that category. Grammatical unification is then implemented simply as term unification in Prolog, which is the implementation language used in the system.

In the semantic interpretation phase, logical form expressions are computed bottom-up by applying semantic interpretation rules keyed to the syntax rules. Terms in the logical form language have semantic sorts associated with them, and functors are restricted with respect to the sorts of their arguments. These sort restrictions are applied as the logical forms are constructed, acting as a filter on the structures produced by the syntactic and semantic rules. The outputs of the semantic interpretation phase are quasi-logical forms in which the scope of quantified noun phrases has not yet been determined. *Quantifier scope is assigned in the next phase of processing.*

At this point in processing, a database-independent formal representation of the meaning of the query has been assigned. This is transformed into a database query, principally by replacing the logical-form constants and predicates derived from the lexicon with database predicates and constants. The query is then re-ordered, if necessary, to optimize database retrieval, and the answer is retrieved from the database, which is stored as a set of Prolog clauses.

Analysis of Results

In the blind test conducted for the June 1990 ATIS evaluation, out of 90 test queries, the SRI system produced correct answers for 25, incorrect answers for 5, and no answer for 60. Thus, the dominant factor in the performance of the SRI system was that most queries failed to get through all stages of processing. Table 1 displays the number and percentage of the queries that failed to get past various levels of processing.

These numbers should be regarded at best as only an approximation of the performance of the different components of the system, for two reasons. First, no attempt has been made to judge the correctness of the output of individual system phases, only to determine whether the phase produced an answer at all. Second, the failure rate of the later phases of processing would probably have been higher if more queries had gotten past the earlier

Level	Number	Percent
Lexicon	1	1.1
Parsing	14	15.5
Interpretation	28	31.1
DB query gen.	17	18.9

Table 1: Analysis of SRI ATIS Results

phases of processing.

With these caveats, the results seem to indicate that most of the difficulties arose in the semantic interpretation phase and the database query generation phase. The grammar seemed to provide fairly good coverage of the syntactic constructions used, and the lexicon performed surprisingly well given that the vocabulary in the test was completely uncontrolled. Undoubtedly, many of the parsing and interpretation failures were due to the absence of some of the necessary lexical entries for particular words, but almost no words in the test material were totally absent from the lexicon.

The semantic rules and the database query generator are, in fact, the parts of the system that are the most recent in origin and must be regarded as far from complete, independently of how they performed on this evaluation. Our main conclusion, then, is simply that much more work is needed on these parts of the system.

Recommendations

In the course of working with the ATIS database and development data, it seemed to the SRI team that there are a number of changes in the database structure and the data collection system that would result in more interesting data being collected, and that would make system development easier for ATIS system builders. The philosophy that Texas Instruments followed in setting up the data collection system was to present information to the subject in a way that mirrored as closely as possible the way the information is presented in the printed Official Airline Guide (OAG). We believe that an attempt should be made to tailor the presentation of information to the capabilities of eventual interactive spoken-language computer systems rather than the printed page. The current ATIS data collection system presents a lot of information to the subject in response to most queries, but does so by using many abbreviated codes and column headings that are compressed in order to fit as much information as possible on one line of the screen. This is appropriate for a printed document, because of the difficulties of cross-referencing mutiple tables in different parts of a printed volume, and because of the need to keep the physical size of the volume down to manageable proportions. Neither of these reasons applies to an interactive spoken language computer system where cross-referencing is easly performed by the system, and much larger volumes of data are easily handled.

We would recommend that the data collection system be modified to present less information in response to most queries, but to present that information in a fuller, less abbreviated form. It has been widely noted that about one-third of the ATIS queries collected so far are about the meaning of codes or abbreviated column headings in the displays, rather than about the domain. If fewer columns were presented in each display, it would be possible to avoid the use of many of these abreviations. Moreover, it might prompt subjects to ask more follow-up questions to retrieve the information not displayed, generating a wider range of queries in the domain of air travel planning.

Implementing this recommendation will require changing not only the displays, but also the structure of the database, so that database tuples that differ only in information not displayed to the subject can be eliminated. Otherwise, the subject would see what appear to be duplicate answers in the display.

A number of other changes to the structure of the database would also seem to be desirable. One significant problem is the status of connecting flights. We believe it is important to devote some thought and attention to restructuring the database to put connecting flights on an equal footing with direct flights in the ATIS database. Currently, these are not even listed in the flight table, so that requests for all flights that meet certain constraints result in information only about direct flights. As a result there are almost no queries about connecting flights in the ATIS data, perhaps because the subjects are not aware of their existence. A related issue is that there is no fare information on connecting flights, because it is not presented in the printed OAG. We believe that if fare information for connecting flights cannot be obtained from OAG, then reasonable fares should be computed for them.

These seem to us to be the most important database and data collection system issues that need to be addressed for future ATIS evaluations, but there are many other smaller issues as well. We therefore suggest that a task force should be created to address these issues and decide on changes to be implemented for future ATIS evaluations.

References

[1] Alshawi, et al, *Interim Report on the SRI Core Language Engine*, Technical Report CCSRC-5, SRI International, Cambridge Research Centre, Cambridge, England, 1988.

Efficient Bottom-Up Parsing

Robert Moore and John Dowding

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Abstract

This paper describes a series of experiments aimed at producing a bottom-up parser that will produce partial parses suitable for use in robust interpretation and still be reasonably efficient. In the course of these experiments, we improved parse times by a factor of 18 over our first attempt, ending with a system that was twice as fast as our previous parser, which relied on strong top-down constraints. The major algorithmic variations we tried are described along with the corresponding performance results.

Introduction

Elsewhere [1] we describe a change in our approach to NL processing to allow for more robust methods of interpretation. One consequence of this change is that it requires a different type of parsing algorithm from the one we have been using. In our previous SLS work, we have used a shift-reduce left-corner parser incorporating strong top-down constraints derived from the left context, to limit the structures built by the parser [2]. With this parser, no structure is built unless it can combine with structures already built to contribute to an analysis of the input as a single complete utterance. If we want to find grammatical fragments of the input that may be of use in robust interpretation, however, such strong use of top-down constraints is not appropriate.

To address this issue, we have built and measured the performance of a number of bottom-up parsers. These parsers use the same unification grammar as our shift-reduce parser, but they do not impose the strong top-down constraints of the original. These experimental parsers fall into two groups: purely bottom-up parsers and bottom-up parsers that use limited top-down constraints. The experiments were performed using a fixed grammar and lexicon for the Air Travel Information System (ATIS) domain, and an arbitrarily selected test corpus of 120 ATIS0 training sentences. The test grammar could produce complete parses for 79 of these 120 sentences.

Pure Bottom-Up Parsing

The first parser we implemented was a straightforward "naive" implementation of the CKY algorithm [3, 4] adapted to unification grammar. In this algorithm, a "chart" is maintained that contains records, or "edges," for each type of linguistic category that has been found between given start and end positions in a sentence. In context-free parsing, these categories are simply the nonterminal symbols of the grammar. In a unification grammar, they are complex structures that assign values to particular features of a more general category type.

Our naive algorithm simply seeds the chart with edges for each possible category for all the words in the sentence, and then works left to right constructing additional edges bottom-up. Each time an edge is added to the chart, the grammar is searched for rules whose last category on the right-hand side matches the edge just added to the chart, and the chart is scanned back to the left for a contiguous sequence of edges that match the remaining categories on the right-hand side of the rule. If these are found, then an edge for the category on the left-hand side of the rule is added to the chart, spanning the segment of the input covered by the sequence of edges that matched the right-hand side of the rule.

When measured with our test grammar and test corpus, our implementation of this algorithm is almost nine times slower than our original shift-reduce parser. We conjectured that one significant problem was the unconstrained hypothesization of empty categories or "gaps." Our grammar, like many others, allows certain linguistic phrase types to be realized as the empty string in order to simplify the overall structure of the grammar. For example, "What cities does American fly to from Boston?" is analyzed as having an empty noun phrase between "to" and "from," so that most of the analysis can be carried out using the same rules that are used to analyze such sentences as "Does American fly to Dallas from Boston?" Because empty categories are not directly indicated in the word string, our naive bottom-up parser must hypothesize every possible empty category at every point in the input.

To address this point, we applied a well-known transformation to the grammar to eliminate empty categories by adding additional rules. For each type of empty cat-

egory, we found every case where it would unify with a category on the right-hand side of a rule, performed the unification, and deleted the unified empty category from the rule. For example, if B can be an empty category then from $A \rightarrow BC$ we would derive the rule $A \rightarrow C$, taking into account the results of unification. When all such derived rules are added to the grammar, all the empty categories can be eliminated.

Performing this transformation both reduced the number of edges being generated and speeded up parsing, but only by about 20 percent in each case. We observed that the elimination of empty categories had resulted in a grammar with many more unit production rules than the original grammar; that is, rules of the form $A \rightarrow B$. This occurred because of the large number of cases like the one sketched above, where an empty category matches one of the categories on the right-hand side of a binary branching rule. We determined that the application of these unit production rules accounted for more than 60 percent of the edges constructed by the parser.

Our next thought, therefore, was to try to transform the grammar to eliminate unit productions as well, but this process turned out to be, in practical terms, intractable. Eliminating empty categories had increased the grammar size but only by about half. When we tried to eliminate unit productions, processing the first four (out of several hundred) grammar rules took a couple of hours of computation time and generated more than 1800 derived rules. We abandoned this approach, and instead we eliminated the unit productions from the grammar by compiling them into a "link table." The link table is basically the transitive closure of the unit productions, so it is, in effect, a specification of the unit derivations permitted by the grammar, omitting the intermediate nodes. This table is then used by the parser to find a path via unit productions between the edges in the chart and the categories that appear in the nonunit grammar rules. This is effectively the same as the CKY algorithm except that edges that would be produced by unit derivations are never explicitly created.

We also made some modifications to speed up selection of applicable grammar rules. We added a "skeletal" chart that keeps track of the sequences of general categories (ignoring features) that occur in the chart (or could be generated using the link table), with the restriction that the only sequences recorded are those that are initial segments of the sequence of general categories (ignoring features) on the right-hand side of some grammar rule. Each grammar rule is itself indexed by the sequence of general categories occuring on its right-hand side. For example, if there is some sort of verb spanning position x through position y in the input and some some sort of noun phrase spanning position y through position z, the skeletal chart would record that there is a sequence of type v_np ending at point z. Thus, when the parser searches for applicable rules to apply to generate new edges in the chart at a particular position, it only considers rules which are indexed by an entry in the skeletal chart for that position.

Eliminating unit productions by use of the link table and accessing the grammar rules through the skeletal chart made the parser substantially faster, but this parser is still almost three times slower than the shift reduce parser on our test corpus using our test grammar. At this point, we seemed to have reached a practical limit to how fast we could make the parser while still constructing essentially every possible edge bottom-up. This parser is in fact almost twice as fast as the shift-reduce parser in terms of time per edge constructed, but it constructs more than four times as many edges.

Making Limited Use of Context

Our limited success in constructing a purely bottom-up parser that would be efficient enough for practical use with our unification grammar led us to reconsider whether it is really necessary to compute every phrase that can be identified bottom-up in order to use the output of the parser in a robust interpretation scheme. We again focused our attention on syntactic gaps. Although we had dealt effectively with explicitly empty categories and with categories generated by the unit productions created by the elimination of empty categories, we knew that many of the additional edges the bottom-up parser was creating were for larger phrases that implicitly contain gaps (e.g., a transitive verb phrase with a missing object noun phrase), even when there is nothing in the preceding context to license such a phrase. We reasoned that there is little benefit to identifying such phrases, the vast majority of which would be spurious anyway, because unless we can determine the semantic filler of a gap, the phrase containing it is unlikely to be of any use in robust interpretation.

With this rationale, we have implemented several variants of a bottom-up parsing algorithm that allows us to use limited top-down constraints derived from the left-context to block the formation of just the phrases that implicitly contain gaps not licensed by the preceding context. For example, in the sentence we previously discussed, "What cities does American fly to from Boston?" the interrogative noun phrase "what cities" signals the possible presence of a noun phrase gap later in the sentence. This licenses

fly to
fly to from Boston
American fly to from Boston
does American fly to from Boston

all as being legitimate phrases that contain a noun phrase gap. Without that preceding context, we would not want to consider any of these word strings as legitimate phrases.

To implement this approach we partitioned the set of grammatical categories into context-independent and context-dependent subsets, with the context-dependent categories being those that implicitly contain gaps. Defining which categories those are is relatively easy in our grammar, because we have a uniform treatment of

"wh" gaps, usually called "gap-threading" [5], so that every category that implicitly or explicitly contains a gap has a feature `gapsin` whose value is something other than `null`. We have a similar treatment of the fronting of auxiliary verbs in yes/no questions, controlled by the feature `vstore`. Finally, an additional quirk of cur grammar required us to treat all relative clauses as context dependent categories. So we defined the context-independent categories to be those that

- Have `null` as the value of `gapsin` or lack the feature `gapsin`, and
- Have `null` as the value of `vstore` or lack the feature `vstore`, and
- Are not relative clauses.

All other categories are context dependent.

This is, of course, simply one of any number of ways that categories could be divided between context-independent and context-dependent. Our ability to change these declarations gives us an interesting parameterization of our parser, such that it can be run as anything from a purely bottom-up parser, if all categories are declared context-independent, to one that uses maximum prediction based on left context like our shift-reduce parser, if all categories are declared context-dependent. It would also be possible to derive a candidate set of context-dependent categories automatically or semi-automatically from a corpus. The candidates for context-dependent categories would be those categories that most often fail to contribute to a complete parse when found bottom-up.[1]

The basic parsing algorithm remains the same as in the purely bottom-up parsers, with a few modifications. After each rule application the resulting category is checked to see whether it unifies with one of the context-independent categories. If so, the edge for it is added to the chart with no further checking. If not, a test is made to see whether the category is predicted by the preceding left context. If so, it is added to the chart; otherwise, it is rejected.

The main complexities of the algorithm are in the generation and testing of predictions. Whenever an edge is added to the chart, predictions are generated that are similar to "dotted rules" or "incomplete edges," except that predictions include only the remaining categories to be matched, since predictions are not used in a reduction step as they are in other algorithms. So, if we have a rule of the form $A \rightarrow BC$ and we add an edge for B to the chart, then we may add a prediction for C following B. Whether the prediction is made or not depends on a number of things, including whether the left-hand side of the rule is context-dependent or independent. In the current example, if A is a context-independent category, we proceed with the prediction; otherwise, we must check whether A itself is predicted. In addition, predictions can arise from matching part of a previous prediction. If we have predicted AB and we find A, then we can predict B.

In order to minimize the number of predictions made, we make two important checks. First we check that the prediction actually predicts some context-dependent category. Second, we do a "follow" check, to make sure that the predicted category might occur, given the next word in the input stream. There are a few other minor refinements to limit the number of predictions, but these are the most important ones. In order to check whether a context-dependent category is predicted by a certain prediction, we consult a "left-corner reachability table" that tells us whether the category we are testing is a possible left corner of the predicted category.

When we tested this algorithm, we found that it dramatically reduced the number of edges generated, and equally dramatically improved parse time. We noted above that our best purely bottom-up parser was about three times slower that the shift-reduce parser. This algorithm proved to be 20 percent faster than the shift-reduce parser on our test corpus and test grammar.

Examination of the number and type of edges produced by this weakly-predictive parser led us to question whether all the refinements that we had made to the purely bottom-up parsers, in order to deal with the enormous number of edges they produced, were still necessary. We have performed a number of experiments removing some of those refinements, with interesting results. The main effect we observed was that using the link table to avoid creating edges for categories produced by unit derivations is no longer productive. By using the link table to create explicit edges for those categories, so that we do not have to use the link table at the time we match the right-hand sides of rules against the chart, we got a parser that was twice as fast as the shift reduce parser. We also found that leaving empty categories in the grammar actually speeded-up this version of the parser very slightly (about 4 percent). More edges and predictions were generated for the empty categories, but this was apparently more than compensated for by the reduction in the number of grammar rules.

Conclusions

This paper is, in effect, a narrative of an exercise in algorithm design and software engineering. Unlike most algorithms papers, it contains a great deal of detail on what did not work, or at least what did not work as well as had been hoped. It is also notable because it talks about practical, rather than theoretical efficiency. Most papers on parsing algorithms focus on theoretical worst-case time bounds. Although we have not analyzed it, it seems likely that all the algorithms we tried have the same polynomial time bound, but the difference in the constants of proportionality involved makes all the difference between the algorithms being usable and not usable. Also, unlike most experimental results on parsing, ours are based on a real grammar, being developed

[1] This idea arose in response to a question posed by Mitch Marcus.

for a real application, not a toy grammar written only for the purposes of testing parsing algorithms. It is unlikely that the problems with gaps that are absolutely crucial in this exercise would arise in such a toy grammar.

In terms of concrete results, the relative performance of several of the parsers is summarized in the table below.

Parser	Time	# Edges	Time/Edge
shift-reduce	1.00	1.00	1.00
naive bottom-up	8.81	12.52	0.70
best bottom-up	2.95	4.62	0.63
best predictive	0.48	1.79	0.27

Notice that all the new parsers are significantly faster than the shift-reduce parser in terms time per edge generated. This is undoubtedly due to the high overhead of the prediction mechanism used in the shift-reduce parser. It is also interesting to note that among the new parsers, the faster the overall speed of the parser, the faster the time per edge, also. This may be somewhat surprising, because of all the additional mechanisms added to the last two parsers to reduce the number of edges, compared to the naive bottom-up parser. Evidently the benefits of having a smaller chart to search outweighed the costs of the additional mechanism, even on the basis of time per edge.

In summary, our first attempt to produce a bottom-up parser was nine times slower than our baseline system; our last attempt was twice as fast. Thus we achieved a speed up of a factor of 18 over the course of these experiments. We finished not only with a parser that produced the additional possible phrases that we wanted for robust interpretation, but did so much faster than the parser we started with. Furthermore, we have developed what seems to be an important new parsing method for grammars that allow gaps, and perhaps more generally for grammars with a set of categories that can be divided into those constrainted mainly internally and those with important external constraints.

Acknowledgments

This research was supported by the Defense Advanced Research Projects Agency under Contract N00014-90-C-0085 with the Office of Naval Research.

References

[1] E. Jackson, D. Appelt, J. Bear, R. Moore, and A. Podlozny, *A Template Matcher for Robust NL Interpretation*, Proceedings, Fourth DARPA Workshop on Speech and Natural Language (February 1991).

[2] R. Moore, D. Appelt, J. Bear, M. Dalrymple, and D. Moran, SRI's Experience with the ATIS Evaluation, Proceedings, DARPA Speech and Natural Language Workshop (June 1990).

[3] T. Kasami, "An Efficient Recognition and Syntax Algorithm for Context-Free Languages," Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, Massachusetts (1965).

[4] D. H. Younger, "Recognition and Parsing of Context-Free Languages in Time n^3," *Information and Control* Vol. 10, No. 2, pp. 189–208 (1967).

[5] L. Karttunnen, "D-PATR: A Development Environment for Unification-Based Grammars," Proceedings of the 11th International Conference on Computational Linguistics, Bonn, West Germany, pp. 74–80 (1986).

DARPA Workshop: Speech & Natural Language
February 1991

SPEECH RECOGNITION IN SRI'S RESOURCE MANAGEMENT AND ATIS SYSTEMS

Hy Murveit, John Butzberger, Mitch Weintraub

SRI International, Menlo Park, CA 94025

OVERVIEW

This paper describes improvements to DECIPHER, the speech recognition component in SRI's Air Travel Information Systems (ATIS) and Resource Management systems. DECIPHER is a speaker-independent continuous speech recognition system based on hidden Markov model (HMM) technology. We show significant performance improvements in DECIPHER due to (1) the addition of tied-mixture HMM modeling (2) rejection of out-of-vocabulary speech and background noise while continuing to recognize speech (3) adapting to the current speaker (4) the implementation of N-gram statistical grammars with DECIPHER. Finally we describe our performance in the February 1991 DARPA Resource Management evaluation (4.8 percent word error) and in the February 1991 DARPA-ATIS speech and SLS evaluations (95 sentences correct, 15 wrong of 140). We show that, for the ATIS evaluation, a well-conceived system integration can be relatively robust to speech recognition errors and to linguistic variability and errors.

Introduction

The DARPA ATIS Spoken Language System (SLS) task represents significant new challenges for speech and natural language technologies. For speech recognition, the SLS task is more difficult than our previous task, DARPA Resource Management, along several dimensions: it is recorded in a noisier environment, the vocabulary is not fixed, and, most important, it is spontaneous speech, which differs significantly from read speech. Spontaneous speech is a significant challenge to speech recognition, since it contains false starts, and non-words, and because it tends to be more casual than read speech. It is also a major challenge to natural language technologies because the structure of spontaneous language differs dramatically from the structure of written language, and almost all natural language research has been focused on written language.

SLS Architecture

SRI has developed a spoken language system (SLS) for DARPA's ATIS benchmark task [1]. This system can be broken up into two distinct components, the speech recognition and natural language components. DECIPHER, the speech recognition component, accepts the speech waveform as input and produces a word list. The word list is processed by the natural language (NL) component, which generates a data base query (or no response). This simple serial integration of speech and natural language processing works well because the speech recognition system uses a statistical language model to improve recognition performance, and because the natural language processing uses a template matching approach that makes it somewhat insensitive to recognition errors. SRI's SLS achieves relatively high performance because the SLS-level system integration acknowledges the imperfect performance of the speech and natural language technologies. Our natural language component is described in another paper in this volume [2]. This paper focuses on the speech recognition system and the evaluation of the speech recognition and overall ATIS SLS systems.

Resource Management Architecture

SRI has also evaluated DECIPHER using DARPA's Resource Management task [3,4]. The system architecture for this task is simply the speech recognition system with no NL postprocessing. There are two language models used in the evaluation: a perplexity 60 word-pair grammar, and a perplexity 1000 all-word grammar. The output is simply an attempted transcription of the input speech.

DECIPHER

This section reviews the structure of the DECIPHER system [5]. The following sections describe changes to DECIPHER.

Front End Analysis

DECIPHER uses an FFT-based Mel-cepstra front end. Twenty-five FFT-Mel filters spanning 100 to 6400 Hz are used to derive 12 Mel-cepstra coefficients every 10-ms frame. Four features are derived every frame from this cepstra sequence. They are

- Energy-normalized Mel-cepstra
- Smoothed 40-ms time derivatives of the Mel-cepstra
- Energy
- Smoothed 40-ms energy differences.

We use 256-word speaker-independent codebooks to vector-quantize the Mel-cepstra and the Mel-cepstral differences. The resulting four-feature-per-frame vector is used as input to the DECIPHER HMM-based speech recognition system.

Pronunciation Models

DECIPHER uses pronunciation models generated by applying a phonological rule set to word baseforms. The techniques used to generate the rules are described in [6] and [5]. These generate approximately 40 pronunciations per word as measured on the DARPA Resource Management vocabulary and 75 per word on the ATIS vocabulary. Speaker-independent pronunciation probabilities are then estimated using these bushy word networks and the

forward-backward algorithm in DECIPHER. The networks are then pruned so that only the likely pronunciations remain—typically about 4 per word for the resource management task and 2.6 per word on the ATIS task. This modeling of pronunciation is one of the ways that DECIPHER is distinguished from other HMM-based systems. We have shown in [6] that this modeling reduces error rate.

Acoustic Modeling

DECIPHER builds and trains word models by using context-dependent phone models arranged according to the pronunciation networks for the word being modeled. Models used include unique-phone-in-word, phone-in-word, triphone, biphone, and generalized biphones and triphones, as well as context-independent models. Similar contexts are automatically smoothed together, if they do not adequately model the training data, according to a deleted-estimation interpolation algorithm similar to [7]. The acoustic models reflect both inter-word and across-word coarticulatory effects. Training proceeds as follows:

- Initially, context-independent boot models are estimated from hand-labels in the TIMIT training database.
- The boot models are used as input for a two-iteration context-independent model training run, where context-independent models are refined and pronunciation probabilities are calculated using the full word networks. These large networks are then pruned by eliminating low probability pronunciations.
- Context-dependent models are then estimated from a second two-iteration forward-backward run, which uses the context-independent models and the pruned networks from the previous iterations as input.

ACOUSTIC MODELING IMPROVEMENTS

Tied Mixtures

We have implemented tied-mixture HMMs (TM-HMMs) in the DECIPHER system. Tied mixtures were first described by Huang[9] and more recently in by Bellegarda and Nahamoo[8]. TM-HMMs use Gaussian mixtures as HMM output probabilities. The mixture weights are unique to each phonetic model used, but the set of Gaussians is shared among the states. The tied Gaussians could be viewed as forming a Gaussian-based VQ codebook that is reestimated by the HMM forward -backward algorithm.

Our implementation of TM-HMMs has the following characteristics:

- We used 12-dimensional diagonal-covariance Gaussians. The variances were estimated and then smoothed with grand variances.
- Computation can be significantly reduced in TM-HMMs by pruning either the mixture weights or the Gaussians themselves. We found that shortfall threshold Gaussian pruning—discarding all Gaussians whose probability density of input at a frame is less than a constant times the best probability density for that frame—works as well for us as standard top-N pruning (keeping the N best Gaussians) and requires less computation.
- We use two separate sets of Gaussian mixtures for our TM-HMMs; one for Mel cepstra and one for Mel-cepstral derivatives. We retained our discrete distribution models for our energy features.
- Corrective training [5,10,11] was used to update the mixture weights for the TM-HMMs. The algorithm is identical to that used for discrete HMMs. That is, the mixture weights are updated as if they were discrete output probabilities. No mixture means or variances were corrected.

We evaluated TM-HMMs on the RM task using the perplexity 60 word-pair grammar. Our training corpus was the standard 3990 sentence training set. We used the combined DARPA 1988, February 1989, and October 1989 test sets for our development set. This contains 900 sentences from 32 speakers. We achieved a 6.8 percent word error rate using our discrete HMM system on this test set. The TM-HMM approach achieved an error rate of 5.5 percent. Thus, the TM-HMMs improved word recognition error rate by 20 percent compared to discrete HMMs.

System Type	Word Error (percent)
Discrete DECIPHER	6.8
Discrete+sex separation	6.3
TM-HMM for recognition only	6.4
TM-HMM	5.5
TM-HMM + sex separation	4.9
TM-HMM + corrective training	4.7
TM-HMM +sex +corrective	4.5

TABLE 1. Error rate improvements with TM-HMMs with our 900-sentence RM development set

Male-Female Separation

In the June 1990 DARPA Speech and Natural Language meeting [5], we reported a 20 percent reduction in RM word-error rate by training separate male and female recognizers, decoding using recognizers from both sexes, and then choosing the sex according to the recognizer with the highest probability hypothesis. This improvement was achieved using a recognizer trained on 11,190 sentences. We did not achieve a significant improvement using male-female separation on the smaller 3990 sentence training set. We set out to see, as has been claimed in [8], whether TM-HMMs can take advantage of male-female separation with smaller (3990 sentence) training sets. Our results were mixed. Although performance did improve from 5.5 percent word error with combined models, to 4.9 percent word error with separate male-female models (a 10 percent improvement) we note that 2/3 of the overall improvement was due to the dramatic improvement for speaker HXS. Aside from this one speaker, the performance gain was not significant. Based on our last study, however, we are confident that male-female separation does improve performance with sufficient training data. The table below shows performance for tied-mixture HMMs using combined and sex-separated models.

Name	Standard Models Errs	Wds	%Err	Male-Female Models Errs	Wds	%Err
ESG	2	241	0.83	4	241	1.66
TAB	4	178	2.25	3	178	1.69
CEW	11	241	4.56	5	241	2.07
AJC	10	253	3.95	6	253	2.37
HXS	36	222	16.22	6	222	2.70
DMS	6	179	3.35	5	179	2.79
GMB	3	246	1.22	7	246	2.85
HLM	11	296	3.72	9	296	3.04
BEF	5	226	2.21	7	226	3.10
TJS	9	265	3.40	9	265	3.40
DAS	14	203	6.90	7	203	3.45
JDH	12	246	4.88	9	246	3.66
EWM	12	272	4.41	10	272	3.68
KLS	8	244	3.28	9	244	3.69
DTD	10	233	4.29	10	233	4.29
AEO	9	229	3.93	10	229	4.37
DML	18	272	6.62	12	272	4.41
PGH	13	204	6.37	9	204	4.41
ERS	11	212	5.19	10	212	4.72
GAW	15	244	6.15	12	244	4.92
AEM	8	302	2.65	17	302	5.63
DTB	7	227	3.08	13	227	5.73
CTW	17	253	6.72	15	253	5.93
CMH	18	230	7.83	15	230	6.52
CRZ	23	302	7.62	20	302	6.62
DWA	19	270	7.04	19	270	7.04
CMR	19	231	8.23	17	231	7.36
JDM	16	271	5.90	21	271	7.75
LNS	21	272	7.72	22	272	8.09
GAG	22	296	7.43	24	296	8.11
JWS	16	222	7.21	21	222	9.46
RKM	22	209	10.53	21	209	10.05
AVG	427	7791	5.48	384	7791	4.93

TABLE 2. Performance with and without sex-separation

There was no significant additional gain from using corrective training in addition to male-female separation. Performance improved from 4.9 percent error (male-female only) or 4.7 percent error (corrective training only) to 4.5 percent error (both methods). This lack of further improvement is due to the reduction in training data.

Speaker Adaptation

We have begun experiments into speaker-adaptation, converting speaker-independent models into speaker-dependent ones. Our experiment involved using VQ codebook adaptation via tied-mixture HMMs as proposed by Rtischev [13]. That is, we adjusted VQ codeword locations based on forward-backward alignments of adaptation sentences. However, since we are using a tied-mixture recognition system, we adapted the Gaussian means instead of the codebook.

We selected 21 of the speakers in our development test set for use in an adaptation experiment. We had either 25 or 30 Resource Management sentences recorded for each of these speakers. We chose to use their first 20 sentences for adaptation, and the other 5 or 10 sentences for adaptation testing.

Using our original TM-HMM models, we achieved an error rate of 7.4 percent (114 errors in 1541 reference words) on this adaptation test set. After adjusting means for each speaker using the 20 adaptation sentences, we achieved an error rate of 6.1 percent (94 errors in 1541 reference words) on the adaptation test sentences.

This improvement with adaptation leads to performance that is still quite short of speaker-dependent accuracy (the ultimate goal of adaptation). Thus, it does not seem worth the added inconvenience of obtaining 20 known sentences from a potential system user, though it is promising for on-line adaptation. We plan to look into several areas for further improvement. For example:

1. Rtischev et al. [14] have shown that adapting mixture weights is at least as important as adapting means.
2. Kubala [15] et al. have shown that adapting speaker-dependent models can be superior to adapting from speaker-independent models.
3. It is possible that the adaptation sentences need not be supervised given the relatively good (7.4 percent error) initial performance.

Rejection of Out-of-Vocabulary Input

We implemented a version of DECIPHER that rejects false input as well as recognizing legal input (our standard recognizer attempts to classify all the input). In addition to standard word models, it uses an out-of-vocabulary word model to recognize the extraneous input. The word model has the following pronunciation network similar to [17].

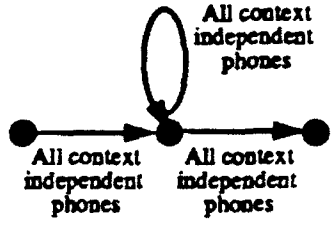


FIGURE 1. Out-of-vocabulary word model

There are 67 phonetic models on each of the arcs in the above word network. All phonetic transition probabilities in this word network are equal, and are scaled by a parameter that adjusts the amount of false rejection vs. false acceptance.

Thus far, we have performed a pilot study that shows this method to be promising. We gathered a database of 58 sentences total from six people. About half of the sentences are digit strings and the other half are digits mixed with other things. There are a total of 426 digits in the database, and 176 additional non-digit words. Example sentences are outlined in Table 3.

We considered correct recognition for these sentences to be the digits in the string without the rest of the words (i.e. 2138767287, 3876541104, 33589170429 are the correct answers for the top three sentences in Table 3).

We trained a digit recognizer with rejection from the Resource Management training set and achieved a word error rate of 5.3 percent for the 27 sentences that contained only digits (13 errors = 1 insert 3 delete 9 subs in 243 reference words), which is within one error of the system without rejection. Thus, in this pilot study, using rejection didn't hurt performance for "clean" input. The overall error rate was 11.7 percent (26 inserts 15 deletes 9 subs in 426 reference words). That is, 402 of 426 digits were detected, and at least 141 of the 176 extraneous words were rejected.

my parents number is 2 1 3 um 8 7 6 ok 7 2 8 7
if you have questions please dial extension 3 8 7 6 at 5 4 1 1 oh 4
please call 3 3 5 8 9 1 um 7 oh 4 2 9
hmm let's see what's this 1 2 3 4 5 uh that's not right 2 3 4 5
1 2 3 oh no that's wrong 2 4 5 8 9 yeah i think that's it
this is a test 1 2 3 4 5 8 7 this was only a test
<grunt> 1 2 <cough> 3 4 5 <sneeze> 8 7 <mic-noise>
4 1 dollars and 3 1 8 cents
what's this oh 4 1 0 8
well let's see 3 1 4 7 8 ok

TABLE 3. Sample sentences for the rejection study

LANGUAGE MODELING

Bigram Language Modeling

We used a bigram language model to constrain the speech recognition system for the ATIS evaluation. A back-off estimation algorithm [16] was used for estimation of the bigram parameters. The training data for the grammar consisted of 5,050 sentences of spontaneous speech from various sites—1,606 from MIT's ATIS data collection project, 774 from NIST CD-ROM releases, 538 from SRI's ATIS data collection project, and 2,132 from various other sites.

Robust estimates for many of the bigram probabilities cannot be achieved since the vast majority of them are seen very infrequently (because of the lack of sufficient training data). Furthermore, frequencies of words such as months and cities were biased by the data collection scenarios and the time of year the data was collected. To reduce these effects, words with effectively similar usage were assigned to groups, and instead of collecting counts for the individual words, counts were collected for the groups. After estimation of the bigram probabilities, the probabilities of transitioning to individual words were assigned the group probability divided by the number of words in the group. This scheme not only reduced some of the problems due to the sparse training data, but also allowed some unseen words (other city names, restriction codes, etc.) to be easily added to the grammar. The table below contains the groups of words tied together.

months, days, digits, teens, decades, date-ordinals, cities, airports, states, airlines, class-codes, restriction-codes, fare-codes, airline-codes, aircraft-codes, airport-codes, other-codes

TABLE 4. Tied Groups

Using our back-off bigram on our ATIS development set (most of the June 1990 DARPA-ATIS test set), we achieved a 14.1 percent word error rate with a test-set perplexity of 19 (not counting 6 words not covered by the grammar). When we applied this grammar to the February 1991 ATIS evaluation test set (200 sentences) the perplexity was 43, excluding 26 instances of words not covered in our vocabulary. For the 148 Class A sentences, the recognition word error rate was 17.8 percent.

We also explored various class-grammar implementations. These grammars were generated by interpolating word-based bigrams with class-based bigrams. We were able to vary the grammars and their perplexities by varying the interpolation coefficients. However, recognition performance never improved over that for the back-off bigram. In fact, accuracy remained relatively constant throughout a large range of perplexities.

Table 5 illustrates recognition accuracy using bigrams with different perplexities on our ATIS development test set. A preliminary set of models was used for recognition (with 442 words in the vocabulary) and the grammars were estimated using 2,909 sentences.

	Perplexity	Word Error (percent)
Backed-off Bigram	19	14.1
Interpolated Bigrams	20	14.5
	24	15.3
	71	14.9
	89	14.7
	91	14.5
	113	14.9
	442	29.2

TABLE 5. Perplexity vs. word error on the ATIS development set

These tables also illustrate that recognition performance did not depend strongly on the test-set perplexity. Clearly, other factors are dominating performance. We believe that one of our most pressing needs in this research is to understand what this bottleneck is, and to develop ways that express it better than perplexity.

Multi-Word Lexical Units

Many words occur with sufficient frequency and with significant cross-word coarticulation that a better acoustic model might be made by training these word combinations as a single word model. These words include "what-are-the," "give-me," etc., which can have a variety of pronunciations best modeled with a network of phones representing the phonetic and phonological variation of the whole sequence ("what're-the," "gimme," etc.) instead of each word separately.

Also, when considering class grammars, multiple word sequences allow classes which could not be constructed by considering every word separately. For instance, having distinct models of all the restriction codes (e.g. "v-u-slash-one") might be more appropriate than modeling *alpha->alpha->slash->number* in the bigram. The latter form would allow all the alphabet letters to transition to all the alphabet letters, with probabilities as prescribed by the bigram, and would incorrectly increase the probability for invalid restriction codes.

This multi-word technique allows all the probabilities of all the restriction codes to be tied together, so that all are equally covered at the appropriate place in the grammar, instead of depending completely on the individual words' statistics estimated from sparse training data. The multi-word approach resulted in only a slight performance improvement compared to a system where non-coarticulatory multi-words were left separated. That is, for the "separate words" system, words like "a p slash eighty" were separate words, but coarticulatory

word models like "what-are-the" and "list-the" were retained. On a 119-sentence subset of the June 90 evaluation set, the results were as shown in Table 6.

Development Set Performance

	Perplexity	Word Error (percent)
Multi-Word	26	9.6
Separate Words	20	10.7

February 1991 Class-A Evaluation Performance

	Perplexity	Word Error (percent)
Multi-Word	43	17.8
Separate Words	34	18.3

TABLE 6. Effectiveness of multi-word modeling

Note that the higher perplexity of the multi-word system is deceiving since high probability grammar transitions are now hidden within the multi-word models, and are not seen by the grammar. Tables 7 and 8 list the various multi-word units.

flights-from, what-is-the, show-me-the, show-me-all, show-me, how-many, one-way, what-are-the, give-me, what-is, i-would-like, i'd-like-to, what-does

TABLE 7. Coarticulatory Multi-Words

CITIES:	san-francisco, washington-d-c, ...
AIRLINES:	a-l, c-o, t-w-a, u-s-air, ...
AIRCRAFT:	d-c-ten, seven-forty-seven, ...
AIRPORTS:	a-t-l, b-o-s, s-f-o, d-f-w, ...
CLASS CODES:	q-x, f-y-b-m-q, k-y, y-n, ...
RESTRICT CODES:	a-p-eighty, a-p-slash-eighty,...
COLUMN HEADS:	d-u-r-a, e-q-p, r-t-a-max, ...

TABLE 8. Semantic Multi-Words

EVALUATION

RM Evaluation

SRI evaluated the DECIPHER system on DARPA's February 1991 speaker-independent test set. The characteristics of the evaluated system were:

- Speaker-independent recognition
- 3990 sentence DARPA-RM training
- 3 state, left-to-right, context-dependent hidden Markov model using deleted-interpolation estimation of parameters
- Input features were 12 Mel-cepstra and delta-Mel-cepstra and scalar quantized energy and delta-energy
- Tied-mixture modeling for Mel cepstra and delta-Mel-cepstra
- 256 diagonal covariance Gaussians for each
- Independent discrete density HMM models for energy and delta energy
- Multiple pronunciation trained phonological modeling, about 4 pronunciations per word on average
- Cross-word acoustic and phonological modeling
- Sex-consistent modeling
- Corrective training on mixture weights
- Resource Management all-word and word-pair grammars used with 992-word Resource Management vocabulary.

We achieved the performance shown in Table 9.

Speaker	P=60	P=1000
ALK03	9.7	20.8
CAL15	2.5	11.9
CAU07	2.6	14.7
EAC02	10.2	22.0
JLS04	1.6	11.1
JWG05	7.5	19.5
MEB03	2.9	17.6
SAS05	2.2	10.4
STK01	4.1	21.2
TBR01	5.2	27.8
Average	**4.8**	**17.6**

TABLE 9. DARPA-RM February 1991 speaker-independent evaluation

Our performance is severely limited by training data[5], and many further improvements for the RM task may only be ways to work around RM's artificial limit on training data. Thus, we expect to develop and evaluate our system in the future with the ATIS task which both has more training data available and uses more realistic (spontaneous) speech.

SLS Evaluation

We evaluated on DARPA's February 1991 ATIS test set using a system similar to the one described above except:

- The system was trained on 17,042 sentences (3990 RM-SI, 4200 TIMIT, 7932 read ATIS, 920 spontaneous ATIS).
- 1,139 word vocabulary (the test set vocabulary was not revealed in advance) using multi-word units.
- Discrete distribution HMM modeling was used for all features.
- A back-off bigram language model [16] with tied word-groups was used, with a test set perplexity of 43 (not counting 26 words out of vocabulary).
- A template-matcher natural language component [2] was used to generate ATIS database queries based on the speech recognition output.

We achieved the performance shown in Table 10.

SPKR	Corr	Sub	Del	Ins	Err	Sent Err
CL	93.6	5.1	1.3	1.7	8.1	42.3
CJ	92.0	6.9	1.0	0.7	8.7	46.2
CO	92.0	3.7	4.3	1.2	9.3	56.2
CP	90.7	7.5	1.8	2.5	11.8	59.3
CK	83.3	8.8	7.8	1.0	17.6	58.3
CH	84.2	5.3	10.5	5.3	21.1	100.0
CE	81.5	12.0	6.5	3.2	21.8	70.0
CI	73.1	24.0	2.9	5.8	32.7	90.0
CM	75.0	23.5	1.5	26.5	51.5	100.0
Average	**86.5**	**10.3**	**3.1**	**4.3**	**17.8**	**60.1**
All-word (Perplexity 1139)						
Average	**86.5**	**23.9**	**3.7**	**8.0**	**35.5**	**91.2**

TABLE 10. DARPA-ATIS February 1991 speech evaluation 148 Class A Sentences

SPKR	Corr	Sub	Del	Ins	Err	Sent Err
CI	91.9	6.5	1.6	0.8	8.9	54.5
CP	91.7	6.6	1.7	1.7	10.0	55.2
CL	91.4	6.7	1.9	1.9	10.4	44.8
CK	85.0	8.7	6.3	0.5	15.5	64.0
CE	83.0	11.8	5.2	2.6	19.6	73.9
CO	79.4	13.7	6.9	1.4	22.0	75.9
CH	78.6	13.1	8.3	3.6	25.0	100.0
CI	67.1	27.3	5.6	5.6	38.6	92.9
CM	72.5	25.2	2.3	23.9	51.4	100.0
Average	83.5	12.6	3.9	4.2	20.7	66.5

TABLE 11. DARPA-ATIS February 1991 speech evaluation All sentences

As can be seen, speakers CI and CM contributed significantly to the overall error rate. Furthermore, many of the errors occurred despite their relatively small bigram probabilities, indicating that the grammar is still not completely effective in overriding poor acoustic matches.

Table 12 describes overall spoken language system performance.

System	Right	Wrong	NA[1]	WErr[2]	Score[3]
NL Only	109	9	27	31.0	69.0
SLS	96	11	38	41.4	58.6

TABLE 12. DARPA-ATIS February 1991 SLS evaluation 148 Class A sentences

Discussion

The most interesting result of this evaluation (see the paper by Pallett in this proceedings) was that, though SRI along with BBN achieved the best speech recognition accuracy, and SRI along with CMU had the best natural-language-only performance, the accuracy of SRI's combined speech and natural language systems was far better than that for the other sites. We attribute this to the error tolerant nature of our speech/natural-language interface. For instance, note that performance using spoken language is not much worse than the performance of the NL component given transcribed input (i.e. given a perfect speech recognition component) even though the SLS speech recognition component had a 60 percent sentence error rate (at least one word was wrong in 60 percent of the sentences).

The above results indicate to us that steady progress in the speech recognition and natural language technologies, together with error-tolerant speech/natural-language interfaces can lead to practical spoken language systems in the near future.

1. NA is no answer

2. WErr or weighted error is percent no answer plus two times the percent wrong.

3. Score = 100 - Werr

REFERENCES

1 Price, P., "The ATIS Common Task: Selection and Overview," *Proceedings DARPA Speech and Natural Language Workshop*, June 1990.

2 Jackson, E., D. Appelt, J. Bear, R. Moore, and A. Podlozny, "A Template Matcher for Robust NL Interpretation," *Proceedings DARPA Speech and Natural Language Workshop*, June 1991.

3 Pallet, D., "Benchmark Tests for DARPA Resource Management Database Performance Evaluations," *Proceedings ICASSP-89*.

4 Price, P., W.M. Fisher, J. Bernstein, and D.S. Pallet, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *Proceedings ICASSP-88*.

5 Murveit, H., M. Weintraub, M. Cohen, "Training Set Issues in SRI's DECIPHER Speech Recognition System," *Proceedings DARPA Speech and Natural Language Workshop*, June 1990.

6 Cohen, M., H. Murveit, J. Bernstein, P. Price, M. Weintraub, "The DECIPHER Speech Recognition System," *Proceedings ICASSP*, April 1990.

7 Jelinek, F. and R. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," pp. 381-397 in E.S. Gelsima and L.N. Kanal (editors), *Pattern Recognition in Practice*, North Holland Publishing Company, Amsterdam, The Netherlands.

8 Bellegarda, J., D. Nahamoo "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. ASSP*, December 1990.

9 Huang, X.D.,"Semi-continuous hidden Markov models for speech recognition," *Computer Speech and Language*, 3 pp. 239-251 (1989)

10 Bahl, L., P. Brown, P. De Souza, and R. Mercer, "A New Algorithm for the Estimation of Hidden Markov Model Parameters," *Proceedings ICASSP-88*.

11 Lee, K-F., and S. Mahajan, *Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition*, Technical Report CMU-CS-89-100, Carnegie Mellon University, January 1989.

12 Huang, X., F. Alleva, S. Hayamizu, H.-W. Hon, M.-Y. Hwang, and K.-F. Lee, "Improved Hidden Markov Modeling for Speaker-Independent Continuous Speech Recognition," *Proceedings DARPA Speech and Natural Language Workshop*, June 1990.

13 Rtischev, Dimitry, *Speaker Adaptation in a Large-Vocabulary Speech Recognition System*, Master's Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

14 Rtischev, D., D.. Nahamoo, and M. Picheny, "Speaker Adaptation via VQ Prototype Modification," submitted to *IEEE Trans. Signal Processing*.

15 Kubala, Francis, Richard Schwartz, and Chris Barry, "Speaker Adaptation from a Speaker Independent Training Corpus," *Proceedings ICASSP-90*.

16 Katz, S., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. ASSP*, March 1987.

17 Asadi, A., R. Schwartz, and J. Makhoul, "Automatic Detection of New Words in a Large Vocabulary Continuous Speech Recognition System," *Proceedings DARPA Speech and Natural Language Workshop*, Oct. 1989.

PERFORMANCE OF SRI'S DECIPHER™ SPEECH RECOGNITION SYSTEM ON DARPA'S CSR TASK

Hy Murveit, John Butzberger, and Mitch Weintraub

SRI International
Speech Research and Technology Program
Menlo Park, CA, 94025

1. ABSTRACT

SRI has ported its DECIPHER™ speech recognition system from DARPA's ATIS domain to DARPA's CSR domain (read and spontaneous *Wall Street Journal* speech). This paper describes what needed to be done to port DECIPHER™, and reports experiments performed with the CSR task.

The system was evaluated on the speaker-independent (SI) portion of DARPA's February 1992 "Dry-Run" WSJ0 test and achieved 17.1% word error without verbalized punctuation (NVP) and 16.6% error with verbalized punctuation (VP). In addition, we increased the amount of training data and reduced the VP error rate to 12.9%. This SI error rate (with a larger amount of training data) equalled the best 600-training-sentence speaker-dependent error rate reported for the February CSR evaluation. Finally, the system was evaluated on the VP data using microphones unknown to the system instead of the training-set's Sennheiser microphone and the error rate only increased to 26.0%.

2. DECIPHER™

The SRI has developed the DECIPHER™ system, an HMM-based speaker-independent, continuous-speech recognition system. Several of DECIPHER™'s attributes are discussed in the references (Butzberger et al., [1]; Murveit et al., [2]). Until recently, DECIPHER™'s application has been limited to DARPA's resource management task (Pallet, [3]; Price et al., [4]), DARPA's ATIS task (Price, [5]), the Texas Instruments continuous-digit recognition task (Leonard, [6]), and other small vocabulary recognition tasks. This paper describes the application of DECIPHER™ to the task of recognizing words from a large-vocabulary corpus composed of primarily read-speech.

3. THE CSR TASK

Doddington [7] gives a detailed description of DARPA's CSR task and corpus. Briefly, the CSR corpus* is composed of recordings of speakers reading passages from the *Wall Street Journal* newspaper. The corpus is divided in many ways; it includes speaker-dependent vs. speaker independent sections and sentences where the users were asked to verbalize the punctuation (VP) vs. those where they were asked not to verbalize the punctuation (NVP). There are also a small number of recordings of spontaneous speech that can be used in development and evaluation.

The corpus and associated development and evaluation materials were designed so that speech recognition systems may be evaluated in an open-vocabulary mode (none of the words used in evaluation are known in advance by the speech recognition system) or in a closed vocabulary mode (all the words in the test sets are given in advance). There are suggested 5,000-word and 20,000-word open- and closed-vocabulary language models that may be used for development and evaluation. This paper discusses a preliminary evaluation of SRI's DECIPHER™ system using read speech from the 5000-word closed-vocabulary tasks with verbalized and nonverbalized punctuation.

4. PORTING DECIPHER™ TO THE CSR TASK

Several types of data are needed to port DECIPHER™ to a new domain:

- A target vocabulary list
- A target language model
- Task-specific training data (optional)
- Pronunciations for all the words in the target vocabulary (mandatory) and for all the words in the training data (optional)
- A backend which converts recognition output to actions in the domain (not applicable to the CSR task).

*The current CSR corpus, designated WSJ0 is a pilot for a large corpus to be collected in the future.

4.1. CSR Vocabulary Lists and Language Models

Doug Paul at Lincoln Laboratories provided us with baseline vocabularies and language models for use in the February 1992 CSR evaluation. This included vocabularies for the closed vocabulary 5,000 and 20,000-word tasks as well as backed-off bigram language models for these tasks. Since we used backed-off bigrams for our ATIS system, it was straightforward to use the Lincoln language models as part of the DECIPHER™-CSR system.

4.2. CSR Pronunciations

SRI maintains a list of words and pronunciations that have associated probabilities automatically estimated (Cohen et al., [8]). However, a significant number of words in the speaker-independent CSR training, development, and (closed vocabulary) test data were outside this list. Because of the tight schedule for the CSR evaluation, SRI looked to Dragon Systems which generously provided SRI and other DARPA contractors with limited use of a pronunciation table for all the words in the CSR task. SRI combined its internal lexicon with portions of the Dragon pronunciation list to generate a pronunciation table for the DECIPHER™-CSR system.

4.3. CSR Training Data

The National Institute of Standards and Technology provided to SRI several CDROMS containing training, development, and evaluation data for the February 1992 DARPA CSR evaluation. The data were recorded at SRI, MIT, and TI. The baseline training conditions for the speaker-independent CSR task include 7240 sentences from 84 speakers, 3,586 sentences from 42 men and 3,654 sentences from 42 women.

5. PRELIMINARY CSR PERFORMANCE

5.1. Development Data

We have partitioned the speaker-independent CSR development data into four portions for the purpose of this study. Each set contains 100 sentences. The respective sets are male and female speakers using verbalized and nonverbalized punctuation. There are 6 male speakers and 4 female speakers in the SI WSJ0 development data.

The next section shows word recognition performance on this development set using 5,000-word, closed-vocabulary language models with verbalized and nonverbalized bigram grammars. The perplexity of the verbalized punctuation sentences in the development set is 90.

5.2. Results for a Simplified System

Our strategy was to implement a system as quickly as possible. Thus we initially implemented a system using four vector-quantized speech features with no cross-word acoustic modeling. Performance of the system on our development set is described in the tables below.

Table 1: Simple Recognizer

Speaker	Verbalized Punctuation %word err	Non Verbalized Punctuation %word err
050	10.0	11.8
053	14.0	17.6
420	14.7	18.1
421	11.9	17.9
051	21.1	18.8
052	20.7	20.2
22g	15.4	19.6
22h	20.8	13.0
422	57.9	40.4
423	15.0	24.6
Average	20.1	20.2

The female speakers are those above the bold line in Table 1. Recognition speed on a Sun Sparcstation-2 was approximately 40 times slower than real time (over 4 minutes/sentence) using a beam search and no fast match (our standard smaller-vocabulary algorithm), although it was dominated by paging time.

A brief analysis of Speaker 422 shows that he speaks much faster than the other speakers which may contribute to the high error rate for his speech.

5.3. Full DECIPHER™-CSR Performance

We then tested a larger DECIPHER™ system on our VP development set. That is, the previous system was extended to model some cross-word acoustics, increased from four to

six spectral features (second derivatives of cepstra and energy were added) and a tied-mixture hidden Markov model (HMM) replaced the vector-quantized HMM above. This resulted in a modest improvement as shown in the Table 2.

Table 2: Full Recognizer

Speaker	Verbalized Punctuation %word err
050	11.1
053	11.7
420	13.7
421	11.0
051	20.0
052	14.2
22g	15.7
22h	14.9
422	48.3
423	13.0
Average	17.4

6. DRY-RUN EVALUATION

Subsequent to the system development, above, we evaluated the "full recognizer" system on the February 1991 Dry-Run evaluation materials for speaker-independent systems. We achieved word error rates of 17.1% without VP and 16.6% error rates with VP as measured by NIST.*

Table 3: Dry-Run Evaluation Results

Speaker	Non Verbalized Punctuation %word err	Verbalized Punctuation %word err
427	9.4	9.0
425	20.1	15.1
z00	14.4	16.7
063	24.5	17.8
426	10.2	10.8
060	17.0	22.9
061	12.3	13.6
22k	25.3	17.6
221	17.8	12.4
424	20.0	18.4
Average	17.1	15.4

7. OTHER MICROPHONE RESULTS

The WSJ0 corpus was collected using two microphones simultaneously recording the talker. One was a Sennheiser HMD-410 and the other was chosen randomly for each speaker from among a large group of microphones. Such

*The NIST error rates differ slightly (insignificantly) from our own measures (17.1% and 16.6%), however, to be consistent with the other error rates reported in this paper, we are using our internally measured error rates in the tables.

dual recordings are available for the training, development, and evaluation materials.

We chose to evaluate our full system on the "other-microphone" data without using other-microphone training data. The error rate increased only 62.3% when evaluating with other-microphone recordings vs. the Sennheiser recordings.

In these tests, we configured our system exactly as for the standard microphone evaluation, except that we used SRI's noise-robust front end (Erell and Weintraub, [9,10]; Murveit, et al., [11]) as the signal processing component.

Table 4 summarizes the "other-microphone" evaluation results. Speaker 424's performance, where the error rate increases 208.2% (from 18.4% to 56.7%) when using a Shure SM91 microphone is a problem for our system. However, the microphone is not the sole source of the problem, since the performance of Speaker 427, with the same microphone, is only degraded 18.9% (from 9.0 to 10.7%). We suspect that the problem is due to a loud buzz in the recordings that is absent from the recordings of other speakers.

8. EXTRA TRAINING DATA

We suspected that the set of training data specified as the baseline for the February 1992 Dry Run Evaluation was insufficient to adequately estimate the parameters of the DECIPHER™ system. The baseline SI training condition contains approximately 7,240 from 84 speakers (half42 male, 42 female).

We used the SI and SD training and development data to train the system to see if performance could be improved with extra data. However, to save time, we used only speech from male speakers to train and test the system. Thus, the training data for the male system was increased from 3586 sentences (42 male speakers) to 9109 sentences (53 male speakers).* The extra training data reduced the error rate by approximately 20% as shown in Table 5.

*The number of speakers did not increase substantially since the bulk of the extra training data was taken from the speaker-dependent portion of the corpus.

Table 4: Verbalized Punctuation Evaluation Results Using "Other Microphones"

Speaker	Microphone	%word error "other mic"	%word error Sennheiser mic	%degradation
427	Shure SM91 desktop	10.7	9.0	18.9
425	Radio Shack Highball	21.4	15.1	41.8
z00	Crown PCC160 desktop	24.9	16.7	49.1
063	Crown PCC160 desktop	29.4	17.8	65.2
426	ATT720 telephone over local phone lines	12.1	10.8	12.0
060	Crown PZM desktop	30.5	22.9	33.2
061	Sony ECM-50PS lavaliere	18.8	13.6	38.2
22k	Sony ECM-55 lavaliere	25.3	17.6	43.8
221	Crown PCC160 desktop	22.8	12.4	83.9
424	Shure SM91 desktop	56.7	18.4	208.2
Average		25.0	15.4	62.3

Table 5: Evaluation Male Speakers with Extra Training Data

Speaker	Baseline Training	Larger-Set Training
060	22.6	15.5
061	13.6	8.2
22k	17.6	16.8
221	12.4	11.3
424	18.4	15.7
426	10.8	9.8
Average	15.8	12.9

Interestingly, this reduced error rate equalled that for speaker-dependent systems trained with 600 sentences per speaker and tested with the same language model used here. However, speaker-dependent systems trained on 2000+ sentences per speaker did perform significantly better than this system.

9. SUMMARY

This is a preliminary report demonstrating that the DECIPHER™ speech recognition system was ported from a 1,000-word task (ATIS) to a large vocabulary (5,000-word) task (DARPA's CSR task). We have achieved word error rates between of 16.6% and 17.1% as measured by NIST on DARPA's February 1992 Dry-Run WSJ0 evaluation where no test words were outside the prescribed vocabulary. We evaluated using alternate microphone data and found that the error rate increased only by 62%. Finally, by increasing the amount of training data, we were able to achieve an error rate that matched the error rates reported for this task from 600 sentence/speaker speaker-dependent systems. This could not have been done without substantial support from the rest of the DARPA community in the form of speech data, pronunciation tables, and language models.

ACKNOWLEDGEMENTS

We gratefully acknowledge support for this work from DARPA through Office of Naval Research Contract N00014-90-C-0085. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the government funding agencies.

We would like to that Doug Paul at Lincoln Laboratories for providing us with the Bigram language models used in this study, and Dragon Systems for providing us with the Dragon pronunciations described above. We would also like to thank the many people at various DARPA sites involved in specifying, collecting, and transcribing the speech corpus used to train, develop, and evaluate the system described.

REFERENCES

1. Butzberger, J., H. Murveit, E. Shriberg, and P. Price, "Modeling Spontaneous Speech Effects in Large Vocabulary Speech Recognition," DARPA SLS Workshop Proceedings, Feb 1992.

2. Murveit, H., J. Butzberger, and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS Systems," DARPA SLS Workshop, February 1991, pp. 94-100.

3. Pallet, D., "Benchmark Tests for DARPA Resource Management Database Performance Evaluations," *IEEE ICASSP 1989*, pp. 536-539.

4. Price, P., W.M. Fisher, J. Bernstein, and D.S. Pallet, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *IEEE ICASSP 1988*, pp. 651-654.

5. Price, P., "Evaluation of SLS: the ATIS Domain," DARPA SLS Workshop, June 1990, pp. 91-95.

6. Leonard, R.G., "A Database for Speaker-Independent Digit Recognition," *IEEE ICASSP 1984*, p. 42.11

7. Doddington, G., "CSR Corpus Development," DARPA SLS Workshop, Feb 1992.

8. Cohen, M., H. Murveit, J. Bernstein, P. Price, and M. Weintraub, "The DECIPHER™ Speech Recognition System," *IEEE ICASSP-90*.

9. Erell, A., and M. Weintraub, "Spectral Estimation for Noise Robust Speech Recognition," DARPA SLS Workshop October 89, pp. 319-324.

10. Erell, A., and M. Weintraub, "Recognition of Noisy Speech: Using Minimum-Mean Log-Spectral Distance Estimation," DARPA SLS Workshop, June 1990, pp. 341-345.

11. Murveit, H., J. Butzberger, and M. Weintraub, "Reduced Channel Dependence for Speech Recognition", DARPA SLS Workshop Proceedings, February 1992.

REDUCED CHANNEL DEPENDENCE FOR SPEECH RECOGNITION

Hy Murveit, John Butzberger, and Mitch Weintraub

SRI International
Speech Research and Technology Program
Menlo Park, CA, 94025

1. ABSTRACT

Speech recognition systems tend to be sensitive to unimportant steady-state variation in speech spectra (i.e. those caused by varying the microphone or channel characteristics). There have been many attempts to solve this problem; however, these techniques are often computationally burdensome, especially for real-time implementation. Recently, Hermansky et al. [1] and Hirsch et al. [2] have suggested a simple technique that removes slow-moving linear channel variation with little adverse effect on speech recognition performance. In this paper we examine this technique, known as RASTA filtering, and evaluate its performance when applied to SRI's DECIPHER™ speech recognition system [3]. We show that RASTA filtering succeeds in reducing DECIPHER™'s dependence on the channel.

2. INTRODUCTION

A number of techniques have been developed to compensate for the effects that varying microphone and channels have on the acoustic signal. Erell and Weintraub [4, 5] have used additive corrections in the filter-bank log energy or cepstral domains based on equalizing the long-term average of the observed filter-bank log energy or cepstral vector to that of the training data. The techniques developed by Rose and Paul [6] and Acero [7] used an iterative technique for estimating the cepstral bias vector that will maximize the likelihood of the input utterance. Nadas et al. [8] used an adaptive linear transformation applied to the input representation, where the adaptation uses the VQ distortion vector with respect to a predefined codebook. VanCompernolle [10] scaled the filter-bank log energies to a specified range using running histograms, and Rohlicek [9] experimented with a number of histogram-based compensation metrics based on equalizing different aspects of the probability distribution.

One important limitation of the above approaches is that they rely on a speech/nonspeech detector. Each of the above approaches computes spectral properties of the input speech sentence and subsequently compensates for the statistical differences with certain properties of the training data. If the input acoustical signal is not segmented by sentence (e.g. open microphone with no push-to-talk button) and there are long periods of silence, the above approaches would not be able to operate without some type of reliable automatic speech-input/sentence-detection mechanism. An automatic sentence-detection mechanism would have considerable difficulty in reliably computing the average speech spectrum if there were many other nonspeech sounds in the environment.

A second class of techniques developed around auditory models (Lyon [11]; Cohen [12]; Seneff [13]; Ghitza [14]). These techniques use various automatic gain control and other auditory-type modeling techniques to output a spectral vector that has been adapted based on the acoustic history. A potential limitation of this approach is that many of these techniques are very computationally intensive.

3. THE RASTA FILTER

RASTA filtering is a high-pass filter applied to a log-spectral representation of speech. It removes slow-moving variations from the log spectrum. The filtering is done on the log-spectral representation so that multiplicative distortions (such as a linear filter) become additive and may be removed with the RASTA filter. A simple RASTA filter may be implemented as follows:

$$y(t) = x(t) - x(t-1) + (C \cdot y(t-1))$$

where x(t), as implemented in DECIPHER™, is a log band-pass energy which is normally used in DECIPHER™ to compute the Mel-cepstral feature vector. Instead, x(t) is replaced by y(t), the high-pass version of x(t), when performing the cepstral transform.

The constant, C, in the above equation defines the time constant of the RASTA filter. It is desirable that C be such

that short-term variations in the log spectra (presumably important parts of the speech signal) are passed by the filter, but slower variations are blocked. We set C = 0.97 so that signals that vary faster than about 1 Hz are passed and those that vary less than once per second tend to be blocked. Figure 1 below plots the characteristic of this filter.

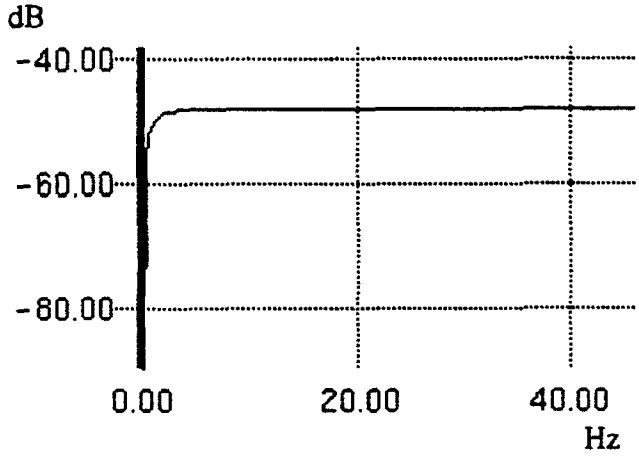


Figure 1: Characteristics of the C = 0.97 RASTA filter

When used in conjunction with SRI's spectral estimation algorithms [4, 5], the high-pass filter is applied to the filter-bank log energies after the spectral estimation operation. The estimates of clean filter-bank energies are highpass filtered and then transformed to obtain the cepstral vector. The cepstral vector is then differenced twice to obtain the delta-cepstral vector and the delta-delta-cepstral vector.

3.1. Removal of an Ideal Linear Filter

We first evaluated RASTA filtering by applying a bandpass filter (Figure 2 below) to a speech recognition task—continuous digit recognition performance over telephone lines. The filter was applied to the test set only (no filtering was applied to the training data). We compared the resulting performance with the performance of an unfiltered test set for both standard and RASTA filtering. As Table 1 shows, the RASTA filtering was successful in removing the effects of the bandpass filter, whereas the standard system suffered a significant performance degradation due to the bandpass filter. Compared with our standard signal processing, the RASTA filtering was able to give a slight improvement on the female digit error rate, with no significant change in the male digit error rate. The dramatic decrease in performance that occurs when the telephone speech is bandpass filtered is removed by the RASTA filtering, and the results are comparable to the original speech signal.

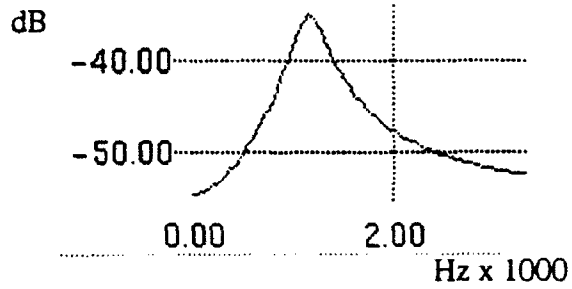


Figure 2: The distorting bandpass filter characteristic.

	Original Speech		Bandpass Speech	
	male	female	male	female
Standard	3.2	3.1	13.9	11.6
RASTA	3.4	2.1	3.0	1.9

Table 1: Word error rates for standard signal processing techniques and RASTA filtering techniques using clean and bandpass-filtered telephone speech.

4. REDUCED MICROPHONE DEPENDENCE

After the encouraging initial study, we tested RASTA filtering in a more realistic manner—measuring the performance improvement, due to RASTA filtering, when dissimilar microphones are used in the test and training data.

To do this, we recorded 50 sentences (352 words) from one talker simultaneously using two different microphones, a Sennheiser flat-response close-talking microphone that was used to train the system, and an Electrovoice 625 handset with a very different frequency characteristic. The user spoke queries for DARPA's ATIS air-travel planning task. Table 2 shows that for this speaker, the error rate was less sensitive to the difference in microphone when RASTA filtering was applied than when it wasn't. Further, there is no evidence from this and the previous study to indicate that RASTA filtering degrades performance when the microphone remains constant.

	Sennheiser	Electro Voice
Standard	13 (3.7%)	31 (8.8%)
RASTA	12 (3.4%)	17 (4.8%)

Table 2: Number and percentage of word errors for a single speaker when test microphone and signal processing were varied.

5. DESKTOP MICROPHONES

RASTA filtering is most effective when differences between training and testing conditions can be modeled as linear filters. However, many distortions do not fit this model. One example is testing with a desktop microphone with models trained with a close-talking microphone. In this scenario, although the microphones characteristics may be approximately related with a linear filter, additive noise picked up by the desktop microphone violates the linear-filter assumption.

To see how important these effects are, we performed recognition experiment on systems trained with sennheiser microphones and tested with a Crown desktop microphone. These test recordings were made at Carnegie Mellon University (CMU) and at the Massachusetts institute of Technology (MIT). They simultaneously recorded a speaker using both Sennheiser and Crown microphones interacting with an ATIS (air travel planning) system.

The performance of DECIPHER™ on the ATIS recordings is shown in Tables 3 and 4. Table 3 shows the system performance results on MIT's recordings, while Table 4 contains the system performance results on CMU's recordings.

Speaker	Sennheiser	Crown	Crown	Crown	Crown
	Standard	Standard	RASTA	NRFE	NRFE+RASTA
4V	13.0	13.8	22.8	18.7	16.3
4W	1.7	5.1	1.7	4.3	3.4
5E	17.8	26.6	27.8	18.1	14.7
55	18.5	26.6	25.3	23.2	17.6
59	13.7	40.2	41.0	26.6	23.6
Average	12.9	22.5	23.7	18.2	15.1

Table 3: Word error rate for MIT recordings varying microphone and signal processing

Speaker	Sennheiser	Crown	Crown	Crown	Crown
	Standard	Standard	RASTA	NRFE	NRFE+RASTA
IF	20.7	91.8	46.9	46.9	36.7
IH	20.5	93.2	75.7	71.0	35.8
IK	26.2	87.1	62.3	60.3	35.8
Average	22.5	90.7	61.6	59.4	36.1

Table 4: Word error rate for CMU recordings varying microphone and signal processing

For the MIT recordings, note that the best performing system on the Crown microphone data was very close with the performance on the Sennheiser recordings (12.9% vs. 15.1%). The addition of RASTA processing did not help the standard processing on the Crown data (the error rate went up slightly from 22.5% to 23.7%) but it did help the noise-robust estimation processing (18.2% to 15.1%).

The performance on CMU's Crown recordings were much lower. CMU's audio recordings for were noticeably noisier; the speaker sounded as if he was much farther from the microphone, and there were other nonstationary sounds in the background. Note that the error rate with the standard signal processing is extremely high (90.7% word error). For the CMU Crown microphone recordings, the addition of RASTA processing helped reduce the error rate for both the standard and noise-robust estimation processing conditions. The NRFE + RASTA processing was able to reduce the error rate by 60% over the no-processing condition on the CMU Crown microphone recordings (90.7% to 36.1%).

SRI's noise-robust spectral estimation algorithms are designed to estimate the filter-bank log energies of the clean speech signal when there is additive colored noise. The estimation algorithms were designed to work independently from any spectral shape introduced by the microphone and channel variations. Therefore, some type of additional spectral normalization is required to compensate for these effects: the combined "NRFE + RASTA" system serves that purpose. The RASTA system (without estimation) can help compensate for the linear microphone effects, but it can help only to a limited degree with the nonlinearities introduced by other sounds.

6. ROBUSTNESS OF REPRESENTATION TO MICROPHONE VARIATION

To understand the benefit that we have obtained using the different processing techniques, we developed a metric for the robustness of the representation that is separate from speech-recognition performance. The DARPA CSR corpus (Doddington [15]) was used for this evaluation since it is contains stereo recordings. By using stereo recordings, we can compare the robustness in the representation that occurs when the microphone is changed. In this CSR corpus, the first channel of these stereo recordings is always a Sennheiser close-talking microphone. The second recording channel uses one of 15 different secondary microphones.

Using this stereo database, we can compute the cepstral feature vector on each microphone channel, and compare the two representations to determine the level of invariance provided by the signal-processing/representation. The metric that we used for determining the robustness of the representation is called relative-distortion and is computed in the following equation.

$$\text{Relative Distortion } (C_i) = \frac{\overline{(C_{i(Mic1)} - C_{i(Mic2)})^2}}{\sigma_{C_{i(Mic1)}} \cdot \sigma_{C_{i(Mic2)}}}$$

The relative distortion for cepstral coefficient C_i is computed by comparing the cepstral value of the first microphone with the same cepstral value computed on the secondary microphone. This average squared difference is then normalized by the variance of this cepstral feature on the two microphones. This metric gives an indication of how much variance there is due to the microphone differences relative to the overall variance of the feature due to phonetic variation. This metric is plotted as a function of the cepstral coefficient for different signal processing algorithms in figure 3.

Figure 3 shows that the RASTA processing helps reduce the distortion in the lower order cepstral coefficients. When combined with SRI's noise-robust spectral estimation algorithms, the distortion decreases even further for the lower order cepstral coefficients. Neither of the algorithms help reduce the distortion for the higher cepstral coefficients. This metric indicates that even though the robust signal processing has reduced the recognition error rate due to microphone differences, there is still considerable variation in the cepstral representation when the microphone is changed.

7. SUMMARY

We have shown that high-pass filtering of the filter-bank log energies can be an effective means of reducing the effects of some microphone and channel variations. We have shown that such filtering can be used in conjunction with our previous estimation techniques to deal with both noise and microphone effects. The high-pass filtering operation is a simple technique that is computationally efficient and has been incorporated into our real-time demonstration system.

REFERENCES

1. H. Hermansky, N. Morgan, A. Bayya, P. Kohn, "Compensation for the Effects of the Communication Channel in Auditory-Like Analysis of Speech," *Eurospeech*, Sept. 1991, pp. 1367-1370.

2. H. Hirsch, P. Meyer, and H.W. Ruehl, "Improved Speech Recognition using High-Pass Filtering of Subband Envelopes," *Eurospeech*, Sept. 1991, pp. 413-416.

3. H. Murveit, J. Butzberger, and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS

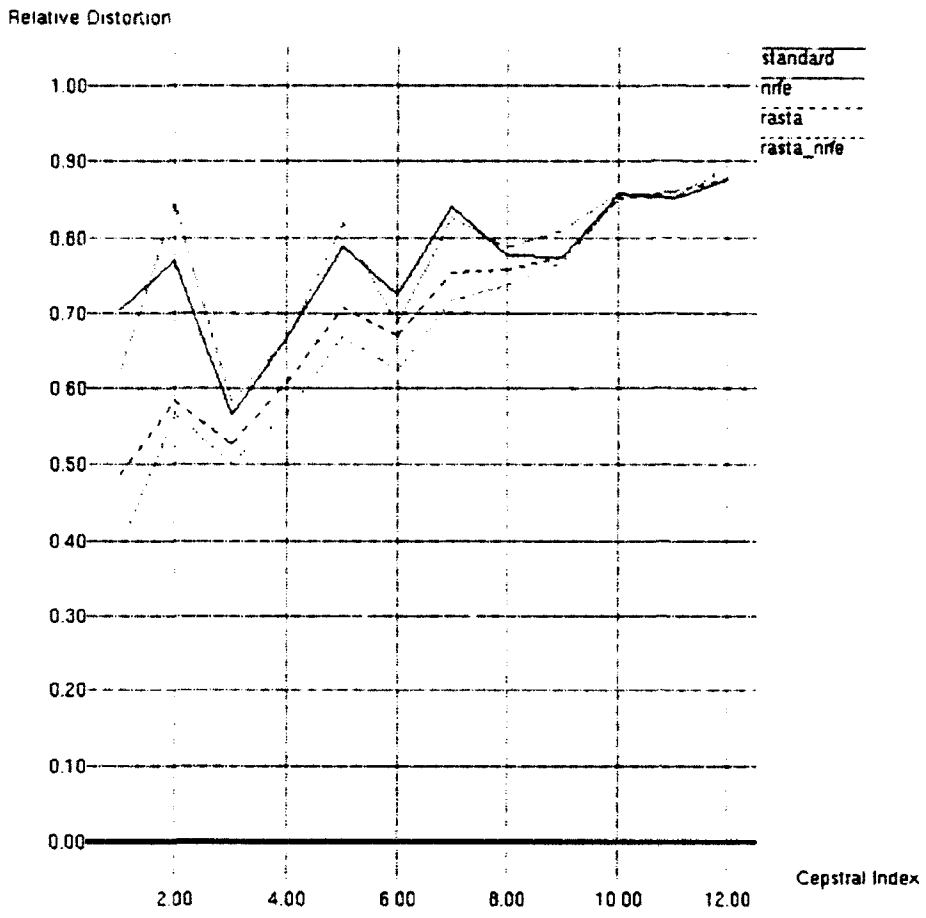


Figure 3: Relative distortion plotted as a function of the cepstral index for different signal processing algorithms (standard, NRFE, RASTA, and RASTA + NRFE).

Systems," DARAP SLS Workshop, February 1991, pp. 94-100.

4. A. Erell, and M. Weintraub, "Spectral Estimation for Noise Robust Speech Recognition," DARPA SLS Workshop October 89, pp. 319-324.

5. A. Erell, and M. Weintraub, "Recognition of Noisy Speech: Using Minimum-Mean Log-Spectral Distance Estimation," DARPA SLS Workshop, June 1990, pp. 341-345.

6. R. Rose and D. Paul, "A Hidden Markov Model Based Keyword Recognition System," *IEEE ICASSP* 1990, pp. 129-132.

7. A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition," Ph.D. Thesis Carnegie-Mellon University, September 1990

8. A. Nadas, D. Nahamoo, M. Picheny, "Adaptive Labeling: Normalization of Speech by Adaptive Transformations based on Vector Quantization" *IEEE ICASSP* 1988, pp. 521-524.

9. R. Rohlicek, W. Russell, S. Roukos, H. Gish, "Continuous Hidden Markov Modeling for Speaker-Independent Word Spotting," *IEEE ICASSP* 1989, pp. 627-630

10. D. VanCompernolle, "Increased Noise Immunity in Large Vocabulary Speech Recognition with the Aid of Spectral Subtraction," *IEEE ICASSP* 1987, pp 1143-1146.

11. R. Lyon, "Analog Implementations of Auditory Models," DARPA SLS Workshop, Feb. 1991 pp. 212-216.

12. J. Cohen, "Application of an Auditory Model to Speech Recognition," *Journ. Acoust. Soc. Amer.*, 1989, 85(6) pp. 2623-2629.

13. S. Seneff, "A Joint Synchrony/Mean Rate Model of Auditory Speech Processing," *Jour. Phonetics*, January 1988

14. O. Ghitza, "Auditory Neural Feedback as a Basis for Speech Processing," 1988 *IEEE ICASSP*, pp 91-94.

15. Doddington, G., "CSR Corpus Development," DARPA SLS Workshop, Feb 1992.

Integrating Natural Language Constraints into HMM-based Speech Recognition

Hy Murveit and Robert Moore
SRI International, Menlo Park, CA

ABSTRACT

This paper discusses a new approach to implementing spoken language systems. This approach both takes full advantage of syntactic and semantic constraints provided by a natural language processing component in the speech understanding task and provides a tractable search space. The results show that the approach is a promising one for large vocabulary systems. We have already achieved, for high perplexity syntactic grammars, parse times within a factor of 20 of real time with resulting HMM recognition computational requirements within the capability of high speed multiprocessor computers or special purpose speech recognition hardware.

INTRODUCTION

This paper discusses a new approach to implementing spoken language systems--systems that integrate speech recognition (SR) and natural language processing (NLP) capabilities. This approach both takes full advantage of syntactic and semantic constraints provided by the NLP and provides a tractable search space for the overall understanding task.

We aim to integrate speech recognition and NLP because:

- many applications of spoken-language systems require understanding of speech, instead of simple recognition,
- appropriate use of constraints from NLP reduces the perplexity of the speech recognition task, increasing word recognition accuracy,
- sharing of information between SR and NLP can improve speech understanding by using acoustic cues to disambiguate certain sentences.

OTHER APPROACHES

Several ways to integrate SR and NLP have been tried. They have the following advantages and disadvantages.

SERIAL CONNECTION BETWEEN SR AND NLP:

In a serial connection, the SR system sends the most likely sentence (based on acoustics) to a NLP system which interprets that sentence.

The advantage of this approach is that the computational burden placed on both the SR and NLP systems is relatively light. The SR system operates as if there are no NL constraints and the NLP system just parses one sentence. The disadvantage is that little interaction between the SR and NLP is possible, i.e. the natural language processor cannot correct errors that the speech recognizer makes.

WORD LATTICE INTERFACE:

The SR system produces a graph representing the recognition scores associated with recognizing all (many) of the words in the vocabulary starting from all (many) possible start times and ending at all (many) possible end times. The NLP system searches this graph for the best scoring sentence that meets NL constraints[1].

The advantage of this approach over the first approach is that it allows interaction between SR and NLP thus improving recognition (and possibly) NLP performance. Disadvantages include a considerably higher computational burden on the system. The SR system must now create a lattice for many word start and end times, and thus may not be able to take advantage of fast dynamic programming based search algorithms appropriate for schemes solving for the best answer only. In the worst case, computation increases by the length of the input sentence (the number of possible start points for every word). Realistically, exhaustive lattices are impractical, and the lattice pruning algorithms that must be used are suboptimal with word lattice interfaces since they cannot make use of the NLP information source. The natural language processor also has much more work as it must evaluate many possible sentences. This is also true for the other approaches below.

N-BEST SENTENCES INTERFACE:

This approach is similar to the serial interface, but the SR system produces the N best sentences instead of the (N=1) best recognized sentence. As with the serial interface, recognizers typically use some language modeling (such as statistical bigrams and trigrams) when determining the top sentences. The NLP system can also produce an NL score for each sentence. It would then choose the sentence with the best combined speech and NLP score. In the case of parse/noparse scoring by the NLP system, the NLP system chooses the first sentence that parses[2].

This approach permits interaction between the SR and NLP components with computation rate increasing linearly with N. Some implementations[2] require that N be known in advance. Researchers using this claim that it runs quicker than a stack-decoder based implementation[3] that generates sentences on demand. In either

case, if N is small the computation rate is low. However, if the correct sentence ranks low in the list of best sentences and NLP can correct this recognition error, then a large N is required and the N-best approach requires more computation than other approaches. Further, as sentence length increases, the N required may increase exponentially.

DYNAMIC NETWORK GENERATION

Our approach, dynamic grammar network generation, represents natural language knowledge in a state transition network, similar to finite-state language models used elsewhere for speech recognition systems. A straight-forward implementation of this approach is not feasible, however, because typical NL systems would generate unmanageably large or infinite networks. Therefore, the network is generated on the fly, and only the portions of the network within a pruned search are expanded. Thus, the state-transition network generated for a particular spoken sentence will be relatively small, and different than that generated for a different utterance.

The approach is described graphically in Figure 1. The system runs as if it were a standard HMM-based speech recognition system using a state-transition network based language model. When the system is started up, the state-transition network contains an initial state, a list of the words that can leave that state (predictions), and markers indicating that the states that would be reached from these initial predictions are blocked--not yet included in the state transition network. The recognition system begins by searching for the words in the initial state's prediction list using a standard beam search. When a state is reached that is not in the network, the SR system calls the NLP system which runs the parser, creates the needed state, and generates predictions for that state. The SR system can then continue until it blocks again. The process of accepting the completion of a word from a state in the network and generating a new state is called a shift, as it corresponds to a shift in a *shift-reduce* natural language parser[4].

This continues until the entire signal is exhausted. Words ending at the end of the signal are checked to see if they reach a final state--a state such that the hypotheses reaching that state are acceptable as complete utterances--and the most probable final-state hypothesis is chosen as the recognized sentence.

This approach allows a tight coupling of SR and NLP algorithms and has the following advantages:

- It brings all knowledge to bear a soon as possible so that extra work need not be done (for instance the recognizer will not pursue hypotheses that can be ruled out by NLP and vice versa). In contrast to an equivalent system based on word lattices, a dynamic-grammar network system would not search portions of the signal that correspond to word-lattice entries that are unlikely due to previous acoustics or natural language.
- It allows for interactions between speech and NLP. For instance, an acoustic recognition model can be altered if the NLP system judges that the word should be emphasized due to its syntactic or semantic position.

In addition, this approach has the important advantage that, from the perspective of the recognition system, finite-state language constraints are used. Thus, all of the experience the speech recognition community has developed for dealing with finite-state-based speech recognition systems still applies to this system. For instance, a standard beam-search pruning technique is used in this system[5].

SYSTEM IMPLEMENTATION

Speech Recognition Processor

This system's speech recognition component is SRI's DECIPHER speech recognition system[6]. It is a continuous speech recognition system that recognizes speech either in a speaker independent or dependent fashion. It uses discrete density 3-state hidden Markov models to represent phones. Four discrete probability densities are used per state to model the variation in vector-quantized Mel-cepstra, quantized derivatives of these Mel-cepstra, quantized energies, and their derivatives. Word models are constructed from network representations of the word pronunciations and from a set of phone models (context-independent, left-biphone, right-biphone, triphone, and unique-phone-in-word models). The system uses a heuristic algorithm to determine which context to use, based on the amount of training data available. However, the more detailed models are smoothed by averaging in less specific models with weights based on an SRI version of IBM's deleted-interpolation algorithm[7].

DECIPHER is routinely used with a finite-state language model (the DARPA word-pair grammar) so converting DECIPHER to be used in this spoken language system was relatively straight-forward.

Natural Language Processor

As we mentioned above, in the dynamic grammar network approach to speech and natural-language generation the NLP incrementally generates a state transition network. We implement this by adapting conventional parsing algorithms, whereby states in the state transition network are used as indices to stored parsing configurations. The parser is called by the recognizer with a state identifier and a word that has been hypothesized by the recognizer starting in that state. The parser looks up the parsing configurations corresponding to the state and attempts to advance each of them by the word hypothesized by the recognizer. (The parsing algorithm incorporates constraints from the left context, so not all words are acceptable in all parsing configurations.) The resulting parsing configurations are stored under a new index, which is passed back to the recognizer as the succeeding state. The NL processor also computes a set of word

predictions for that state.

When a set of parsing configurations is generated, the NL processor can check whether that set of parsing configurations has been generated before, and if so, it will pass back the state identifier that was previously associated with those parsing configurations, to avoid unnecessarily expanding the recognition search space. In practice we have found that it is sufficient to simply let sequences of words that have the same possible grammatical categories lead to the state, as other situations where sets of parsing configurations are duplicated are extremely rare.

Further details of the NL processor implementation are discussed below.

Experimental Evaluations of the Architecture

Speeding Up NL Processing

Initially, we focused on efficient implementations of the NL processor. Previous attempts at parsers for spoken language systems had reported parse times of one or more hours per sentence. Our first results showed that these times could be improved substantially, though those parse times were still far slower than real time. Table 1 shows parse times for a set of 24 sentences tested in a speaker independent system using a perplexity-510 syntax-only NL grammar for a 885 word subset of DARPA's resource management task[8]. No word sequence probabilities are used. The parser runs in Prolog on a SUN4/280 computer. For comparison, a perplexity 991 all-word grammar achieved 82.6% word correct and a perplexity 60 word pair grammar achieved 97.1% correct for this test set.

Mean Sentence Length	Mean Parsing Time (sec)	Mean Active Words per Frame	Cumulative Word Accuracy
7.1	131	1470	88.4%

Table 1.
Parse Times for a Predicting Stack-Based Parser

We next sped up NL processing substantially in two different ways. First, we noticed that a very large proportion of the total NL processing time was consumed generating word predictions for the recognizer; the perplexity of the NL grammar was so high, however, that this resulted in only a modest amount of reduced work done by the recognizer. We therefore altered the interaction between the recognizer and the NL processor, to eliminate the need to compute prediction lists in the NL processor. In the modified architecture, the NL processor "predicts" the entire vocabulary in every state. This has the result that sometimes the recognizer hypothesizes a word that is not a possible continuation of a state and the parser finds that no parsing configurations in that state can be advanced by the word. In this case, the NL processor asks recognizer to prune that hypothesis from further consideration.

The second modification we made was wholly internal to the parser. Most parsing algorithms are position-based, in that they try to find phrases covering particular segments of the input. Since in the dynamic grammar network architecture, the parser does not have access to information about locations of word hypotheses in the input signal, for our initial implementation we chose a stack-based parsing algorithm that did not require input position information. With this algorithm, a parsing configuration was taken to be a stack of grammatical categories corresponding to a partial analysis of an initial segment of the input signal. Later, we realized that it was possible to implement a position-based parser, where the states in the state transition network played the role of input positions. In this parser, the data structures the parser must keep track of are associated directly with states, rather than with stacks. Since, in general, one state corresponds to many stacks, this parser builds many fewer of these data structures, with a resulting increase in efficiency.

The new parser was evaluated with the same 24 sentence test set, with the results shown in Table 2.

Mean Sentence Length	Mean Parsing Time (sec)	Mean Active Words per Frame	Cumulative Word Accuracy
7.1	12	2043	88.4%

Table 2.
Parse Times for a Filtering State-Based Parser

Note that the NLP times are reduced by an order of magnitude, although the mean number of words/frame being evaluated by the HMM system are increased as expected. Eliminating prediction sped up the parser by a factor of 4.9 and using a state-based parser improved the speed by a factor of 2.25.

Extensions to the Grammar

The grammar used for the parsers discussed above parsed only 36% of the sentences in the resource management task. After these experiments were completed, the grammar was extended so that it covered the full 991-Word vocabulary and parsed 91% of the resource management sentences and 85% of the sentences in an independently collected resource management database (a portion of DARPA's TONE database with out-of-vocabulary items modified). Parse times increased with this new grammar by a factor of 3.5.

Fast Match

A fast match was added to the system. Every frame, it uses acoustic evidence to rule out starting up about 80% of the words in the vocabulary without introducing additional error. Thus, even without a predicting parser a particular state at a given time will only have about 200 instead of 991 predictions.

Grammatical State-Width Pruning

The efficiency of our spoken language system architecture is sensitive to the amount of pruning that can be done by the HMM system. If the system hardly prunes, there will be an exponential increase in the number of active words the system has to process as time moves on. However, the system is well behaved with reasonable pruning. Because of this we found that, although most sentences parse normally, there are some sentences that require excessive computation. We found that the key factor in controlling the computation rate was limiting the maximum number of shifts in a sentence. Therefore, we devised the following algorithm that is very similar to the standard HMM-based beam search technique.

- During every frame the grammatical states are sorted by the best internal HMM-scores of each of the state's predictions.
- The system only shifts completed words if the word's predecessor grammatical state is one of the N best states for that frame.

That is, we keep a beam of grammatical states, and only perform shifts for states in the beam. Typical beam sizes used are 20 or 30.

System Evaluation

The system has been evaluated on a portion of the DARPA speaker dependent resource management task. The results in Table 3 are for three speakers using 279 of the 300 development sentences for those speakers (the other 21 sentences didn't parse).

The results in Table 3 indicate that our approach is a promising one for large vocabulary spoken language systems. We have already achieved, for high perplexity syntactic grammars, parse times within a factor of 20 of real time with resulting HMM recognition computational requirements (2500 active words/frame) that are within the capability of high speed multiprocessor computers or special purpose speech recognition hardware.

Our future plans include evaluating other search strategies (e.g. stack decoding), improving the fast match capability, and precompiling more of the run-time computations the parser must perform. We will also incorporate selectional restrictions and word sequence probabilities into our grammar to reduce the resulting perplexity and improve recognition performance. Finally we expect evaluate recognition techniques that model the interactions of natural language, prosody, and phonology, in the context of our tight integration scheme.

REFERENCES

[1] Chow, Y.L., and S. Roukos, "Speech Understanding Using a Unification Grammar," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 727-730, May, 1989.

[2] Schwartz, Richard, and Yen-Lu Chow, "The Optimal N-Best Algorithm: An Efficient Procedure for Finding the Top N Sentence Hypotheses," *Proceedings of the DARPA Speech and Natural Language Workshop*, October, 1989.

[3] Paul, Douglas B., "A CSR/NLP Interface Specification," *Proceedings of the DARPA Speech and Natural Language Workshop*, October, 1989.

[4] Aho, Alfred V., and Jeffrey D. Ullman, *Principles of Compiler Design*, Addison-Wesley, Reading Massachusetts, 1979.

[5] Lowerre, B.T., *The Harpy Speech Recognition System*, PhD Thesis, Comp. Science Dept., Carnegie Mellon University, 1976.

[6] Weintraub, M., Murveit,H., Cohen,M., Price,P., Bernstein,J., Baldwin,G., and Bell,D., "Linguistic Constraints in Hidden Markov Model Based Speech Recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May, 1989.

[7] Jelinek, F., and R. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," pp. 381-397 in *Pattern Recognition in Practice* by E.S. Gelsema and L. N. Kanal (editors), North-Holland Publishing Company, Amsterdam, The Netherlands, 1980.

[8] Price, P., Fisher, W.M., Bernstein, J. and Pallet, D.S., "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April, 1988.

Speaker	Beam	Mean Sentence Length	Mean Parsing Time (sec)	Mean Active Words per Frame	SLS Word Accuracy	P-1000 Word Accuracy
dtb	20	8.5	51	2801	84.7%	77.3%
pgh	20	8.1	49	2407	87.4%	83.6%
rkm	20	8.1	80	2260	79.1%	73.2%

Table 3.
Parse Times for Complete System

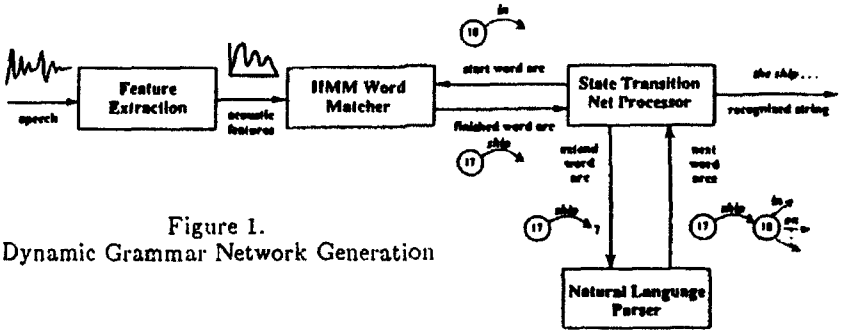


Figure 1.
Dynamic Grammar Network Generation

DARPA Workshop:
Speech & Natural Langua
June 1990

Training Set Issues in SRI's DECIPHER Speech Recognition System

Hy Murveit, Mitch Weintraub, Mike Cohen

Speech Research Program
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025

Abstract

SRI has developed the DECIPHER system, a hidden Markov model (HMM) based continuous speech recognition system typically used in a speaker-independent manner. Initially we review the DECIPHER system, then we show that DECIPHER's speaker-independent performance improved by 20% when the standard 3990-sentence speaker-independent test set was augmented with training data from the 7200-sentence resource management speaker-dependent training sentences. We show a further improvement of over 20% when a version of corrective training was implemented. Finally we show improvement using parallel male- and female-trained models in DECIPHER. The word-error rate when all three improvements were combined was 3.7% on DARPA's February 1989 speaker-independent test set using the standard perplexity 60 wordpair grammar.

System Description

Front End Analysis

Decipher uses a FFT-based Mel-cepstra front end. Twenty-five FFT-Mel filters spanning 100 to 6400 hz are used to derive 12 Mel-cepstra coefficients every 10-ms frame. Four features are derived every frame from this cepstra sequence. They are:

- Vector-quantized energy-normalized Mel-cepstra
- Vector-quantized smoothed 40-ms time derivatives of the Mel-cepstra
- Energy
- Smoothed 40-ms energy differences

We use 256-word speaker-independent codebooks to vector-quantize the Mel-cepstra and the Mel-cepstral differences. The resulting four-feature-per-frame vector is used as input to the DECIPHER HMM-based speech recognition system.

Pronunciation Models

DECIPHER uses pronunciation models generated by applying a phonological rule set to word baseforms. The technique used to generate the rules are described in Murveit89 and Cohen90. These generate approximately 40 pronunciations per word as measured on the DARPA resource management vocabulary. Speaker-independent pronunciation probabilities are then estimated using these bushy word networks and the forward-backward algorithm in DECIPHER. The networks are then pruned so that only the likely pronunciations remain--typically about four pronunciations per word for the resource management task.

This modeling of pronunciation is one of the ways that DECIPHER is distinguished from other HMM-based systems. We have shown in Cohen90 that this modeling improves system performance.

Acoustic Modeling

DECIPHER builds and trains word models by using context-based phone models arranged according to the pronunciation networks for the word being modeled. Models used include unique-phone-in-word, phone-in-word, triphone, biphone, and generalized-phone forms of biphones and triphones, as well as context-independent models. Similar contexts are automatically smoothed together, if they do not adequately model the training data, according to a deleted-estimation interpolation algorithm developed at SRI (similar to Jelinek80). The acoustic models reflect both inter-word and across-word coarticulatory effects.

Training proceeds as follows:

- Initially, context-independent boot models are estimated from hand-labeled portions of the training part of the TIMIT database.
- The boot models are used as input for a 2-iteration context-independent model training run, where context-independent models are refined and pronunciation probabilities are calculated using the large 40-pronunciation word networks. As stated above, these large networks are then pruned to about four pronunciations per word.

- Context-dependent models are then estimated from a second 2-iteration forward-backward run, which uses the context-independent models and the pruned networks as input.

System Evaluation

DECIPHER has been evaluated on the speaker-independent continuous-speech DARPA resource management test sets [Price88] [Pallet89]. DECIPHER was evaluated on the November 1989 test set (evaluated by SRI in March 1990) and had 6% word error on the perplexity 60 task. This performance was equal to the best previously reported error rate for that condition. We recently evaluated on the June 1990 task, and achieved 6.5% word error for a system trained on 3990 sentences and 4.8% word error using 11,190 training sentences.

Since the October 1989 evaluation, DECIPHER's performance has improved in three ways:

- We noted when using that the standard 3990-sentence resource management training set, that many of DECIPHER's probability distributions were poorly estimated. Therefore, we evaluated DECIPHER with several different amounts of training data. The largest training set we used, an 11,190-sentence resource management training set, improved the word error rate by about 20%.
- We implemented a modified version of IBM's corrective training algorithm, additionally improving the word error rate by about 20%.
- We separated the male and female training data, estimated different HMM output distributions for each sex. This also improved word accuracy by 20%.

These improvements are described in more detail below.

Effects of Training Data

In a recent study, we discovered that DECIPHER's word error rate on its training set using the perplexity 60 grammar was very low (0.7% over the 3990 resource management sentences). Since the test-set error rate for that system was about 7%, we concluded that the system would profit from more training data. To test this, we evaluated the system with four databases easily available to us as is shown in Table 1. There *SI* refers to the 3990-sentence speaker-independent portion of the resource management (RM) database--109 speakers, 30 or 40 sentences each, *SD* refers to the speaker-dependent portion of that database--12 speakers, 600 sentences each, and *TIMIT* refers to the training portion of the TIMIT database--420 speakers, 8 sentences each. Note that all SI and SD sentences are related to the resource management task, while TIMIT's sentences are not related to that task. All systems were tested using a continuous-speech, speaker-independent condition with the perplexity 60 resource management grammar testing on DARPA's 300-sentence February 1989 speaker-independent test set.

Training data	Sentences	Word error
SD	7200	7.3
SI	3990	6.7
SI+TIMIT	7350	5.8
SI+SD	11190	5.3

Table 1.
Word Error as a Function of Training Set

Table 1 shows that performance improved as data increased, even when adding the out-of-task TIMIT data. The only exception was that training with 3990 sentences from 100 talkers was slightly better than 7200 sentences from 12 talkers. This is to be expected in a speaker-independent system. This last result is consistent with the findings in Kubala90 that showed that there was not a big performance drop when the number of speakers was drastically reduced (from 109 to 12) in speaker-independent systems. It is likely that more training data would continue to improve performance on this task; however, we believe that a more sensible study would be to focus on how large training sets could improve performance across tasks and vocabularies. (See, for instance, Hon90.)

Separating Male and Female Models

We experimented with maintaining sex consistency in DECIPHER's hypotheses by partitioning male and female training data and using parallel recognition systems as in Bush87. Two *subrecognizers* are run in parallel on unknown speech and the hypothesis from either recognizer with the highest probability is used. The disadvantage of this approach is that it makes inefficient use of training data. That is, in the best scenario the male models are trained from only half of the training data and the female models use only half. This is inefficient because even though there may be a fundamental difference between the two types of speech, they still have many things in common and could profit from the others' training data if used properly.

It is no wonder, then, that this approach has been successful in digit recognition systems with an abundance of training data for each parameter to be estimated, but has not significantly improved performance in large-vocabulary systems with a relatively small amount of training data [Paul89]. To validate the idea of sex consistency, we trained male-only and female-only versions of the DECIPHER speech recognition system using the 11190-sentence SI+SD training set to make sure the data partitions had enough data. We produced SI+SD

subsets with 4160 female and 7030 male sentences. These systems were tested on the DARPA February 1989 speaker-independent test set using the DARPA word-pair grammar (perplexity 60) and are compared below to a similar recognition system trained on all 11190 sentences.

	Standard	Male/Female
Male speakers	5.5	4.6
Female speakers	4.9	4.0
All speakers	5.3	4.3

Table 2. Speaker-Independent %Word Error for Male/Female Parallel Recognizers (February 1989 SI Test Set)

The results in Table 2 show a 19% reduction in the error rate when using sex-consistent recognition systems. This is a significant error rate reduction. A closer look at the system's performance showed that it correctly assigned the talker's sex in each of the 300 test sentences.

Discriminative Techniques Currently in DECIPHER

We have implemented a type of corrective training [Bahl88, Lee89] in the DECIPHER system. Our implementation is similar to that described in Lee89 with the following exceptions or notes:

1. We use four partitions (rather than two) for our deleted estimation technique. In this way, the recognition systems used to generate alignments for corrective training are as similar as possible to the overall recognition system.
2. We do not alter the actual HMM counts for states, but rather scale the states' vector output probabilities by the ratio (#correct+#deletions-#insertions) divided by #correct. These counts are generated by frame alignments of the recognizer hypothesis and the correct sentence. This improves performance from 5.9% word error to 5.1% on the February 1989 test set using the standard SI training set--the uncorrected system has 6.7% word error. The reason for this improvement may be that adjusting the counts of a model affects other models (given our deleted interpolation estimation smoothing algorithms) that do not require correction. Scaling model probabilities only adjusts the models that require change.
3. We do not generate reinforcement errors. We plan to do so using an N-best algorithm to generate alternate hypotheses.
4. We can not iterate the algorithm until the N-best reinforcement is implemented, because the second iteration error rate on the sentences that had been corrected by the first iteration was under 0.3%.

Our implementation reduced the error rate on the February 1989 test set by 24% (6.7% to 5.1%) which is approximately the improvement gained by Lee89 and Bahl88.

Points 3 and 4 above are a concern, because they limit the efficiency with which this algorithm could use its already limited training data. To examine this, we performed the following two experiments. (1) We added a second pass of corrective training, using the speaker-dependent RM training sentences (SD). (2) We combined SD and the SI sentences, thereby using a larger overall training set, but continued to use one pass of corrective training. Table 3 shows that, not surprisingly, though

System	Training	Word Error
no correction	SI	6.7%
1 pass correction	SI	5.1%
add 2nd SD pass	SI	4.6%
no correction	SI+SD	5.3%
1 pass correction	SI+SD	4.1%

Table 3. Corrective Training with Extra Data (Uses February 1989 RM Test Set)

there was improvement when extra data were used as a second pass for the corrective training algorithm, it was better to use these data to simply augment the training data (4.6% versus 4.1% word error). It is also interesting to note that the improvement gained by corrective training with the 3990 SI sentences (6.7% to 5.1%, 24% fewer errors) was approximately equal to the improvement gained by applying corrective training to the larger 11190 SI+SD sentences (5.3% to 4.1%, 23% fewer errors). This leads us to believe that lack of training data is not more of a bottleneck for corrective training than it is for the system as a whole.

Combining Corrective Training and Sex Consistency

We combined both sex consistency and corrective training and arrived at the improvement shown in Table 4. We didn't achieve the same 20% improvement as in the past, probably due to training data limitations.

Attempting the combined system with the standard 3990-sentence training set resulted in poor performance, primarily because the female models used to train

the corrective training partitions had only 870 sentences of training data.

System	Training Data	Word error
Standard	SI	6.7
Standard	SI+SD	5.3
+disc	SI	5.1
+sex	SI+SD	5.3
+disc	SI+SD	4.1
+disc+sex	SI+SD	3.7

Table 4. Summary of Improvements for DECIPHER (Uses February 1989 RM Test Set)

Summary

We have shown significant improvements for the DECIPHER speech recognition system by (1) increasing training data size, (2) implementing corrective training, and (3) separating male and female training data. We have combined all three improvements to achieve our best performing system, one that has a word-error rate of 3.7% on DARPA's resource management February 1989 speaker-independent test set.

We believe that the use of a large training set allows significant improvements in speech recognition accuracy, and therefore we advocate using the larger training set as a standard in future system evaluations.

Acknowledgments

This research was funded by DARPA under the Office of Naval Research contract N00014-90-C-0085 and under the NASA contract NAS2-13120 and under SRI International internal research and development funds.

References

[Bahl88] Bahl, L.R., P.F. Brown, P.V. De Souza, R.L. Mercer, "A New Algorithm for the Estimation of Hidden Markov Model Parameters," *Proceedings ICASSP-88.*

[Bush87] Bush, Marcia A., and Gary E. Kopec, "Network-Based Connected Speech Recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, October 1987

[Cohen90] Cohen, Michael, Hy Murveit, Jared Bernstein, Patti Price, and Mitch Weintraub, "The DECIPHER Speech Recognition System," *Proceedings ICASSP-90.*

[Hon90] Hon, Hsiao-Wuen, and Kai-Fu Lee, "On Vocabulary-Independent Speech Modeling," *Proceedings ICASSP-90.*

[Jelinek80] Jelinek. F. and R. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," pp. 381-397 in E.S. Gelsima and L.N. Kanal (editors), *Pattern Recognition in Practice*, North Holland Publishing Company, Amsterdam, the Netherlands.

[Kubala90] Kubala, Francis, Richard Schwartz, and Chris Barry, "Speaker Adaptation from a Speaker Independent Training Corpus," *Proceedings ICASSP-90.*

[Lee89] Lee, K.F., and S. Mahajan, "Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition," Technical Report CMU-CS-89-100, Carnegie Mellon University, January 1989.

[Murveit89] Murveit, Hy , M. Cohen, P. Price, G. Baldwin, M. Weintraub, and J. Bernstein, "SRI's DECIPHER System," *Proceedings of the DARPA Speech and Natural Language Workshop*, February, 1989.

[Pallet89] Pallet, D., Benchmark Tests for DARPA Resource Management Database Performance Evaluations," *Proceedings ICASSP-89.*

[Paul89] Paul, Douglas, "The Lincoln Continuous Speech Recognition System: Recent Developments and Results," *Proceedings of the DARPA Speech and Natural Language Workshop*, February, 1989.

[Price88] Price, P., W.M. Fisher, J. Bernstein, and D.S. Pallet, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *Proceedings ICASSP-88.*

Evaluation of Spoken Language Systems: the ATIS Domain

P. J. Price

SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025

Abstract

Progress can be measured and encouraged via standards for comparison and evaluation. Though qualitative assessments can be useful in initial stages, quantifiable measures of systems under the same conditions are essential for comparing results and assessing claims. This paper will address the emerging standards for evaluation of spoken language systems.

Introduction and Background

Numbers are meaningless unless it is clear where they come from. The evaluation of any technology is greatly enhanced in usefulness if accompanied by documented standards for assessment. There has been a growing appreciation in the speech recognition community of the importance of standards for reporting performance. The availability of standard databases and protocols for evaluation has been an important component in progress in the field and in the sharing of new ideas. Progress toward evaluating spoken language systems, like the technology itself, is beginning to emerge. This paper presents some background on the problem and outlines the issues and initial experiments in evaluating spoken language systems in the "common" task domain, known as ATIS (Air Travel Information Service).

The speech recognition community has reached agreement on some standards for evaluating speech recognition systems, and is beginning to evolve a mechanism for revising these standards as the needs of the community change (e.g., as new systems require new kinds of data, as new system capabilities emerge, or as refinements in existing methods develop). A protocol for testing speaker-dependent and speaker-independent speech recognition systems on read speech with a 1000-word vocabulary, (e.g., [6]), coordinated through the National Institute of Standards and Technology (NIST), has been operating for several years. This mechanism has inspired a healthy environment of competitive cooperation, and has led to documented major performance improvements and has increased the sharing of methodologies and of data.

Evaluation of natural language (NL) understanding is more difficult than recognition because (1) the phenomena of interest occur less frequently (a given corpus contains more phones and words than syntactic or semantic phenomena), (2) semantics is far more domain dependent than phonetics or phonology, hence changing domains is more labor intensive, and (3) there is less agreement on what constitutes the "correct" analysis. However, MUCK, Message Understanding Conference, is planning the third in a series of message understanding evaluations for later this year (August 1990). The objective is to carry out evaluations of text interpretation systems. The previous evaluation, carried out in March-June 1989, yielded quantitative measures of performance for eight natural language processing systems [4, 5]. The systems are evaluated on performance on a template-filling task and scored on measures of completeness and precision [7].

So far, we have discussed the evaluation of automatic speech recognition (i.e., the algorithmic translation from human speech to machine readable text), and of some aspects of natural language understanding (i.e., the automatic computation of a meaning and the generation, if needed, of an appropriate response). The evaluation of spoken language systems represents a big step beyond the previous evaluation mechanisms described. The input is spontaneous, rather than read, speech. The speech is recorded in an office environment, rather than in a sound-isolated booth. The subjects are involved in problem-solving scenarios. The systems to be tested will be evaluated on the answers returned from a common database. The rest of this paper focuses on the steps taken by the DARPA speech and natural language community to develop a common evaluation database and scoring software and protocols. The first use of this mechanism took place June 1990. However, given the greatly increased challenge, the first use of the mechanism is more a test of the mechanism than of the systems evaluated.

It has become clear in carrying out the evaluation mechanism that the needs of common evaluation are sometimes at odds with the needs of well-designed systems. In particular, the common evaluation ignores dialogue beyond a single query-response pair, and all interactive aspects of systems. A proposal for dialogue evaluation is included in [3], this volume.

Though the initial evaluation mechanism, described below, represents a major effort, and an enormous ad-

vance over past evaluations, we still fall short of a completely adequate evaluation mechanism for spoken language systems. Some forms of evaluation may have to be postponed to the system level and measured in terms of time to complete a task, or units sold. We need to continue to elaborate methods of evaluation that are meaningful. Numbers alone are insufficient. We need to find ways of gaining insight into differences that distinguish various systems or system configurations.

Issues

In this section we will outline the major evaluation issues that have taken up a good deal of our time and energy over the past several months, including: the separation of training and testing materials, black box vs. glass box evaluations, quantitative vs. qualitative evaluation, the selection of a domain, the collection of the data, transcribing and processing the data, documenting and classifying the data, obtaining canonical answers, and scoring of answers.

Independent Training and Test Sets

The importance of independent training/development data and testing data has been acknowledged in speech recognition evaluation for some time. The idea is less prominent in natural language understanding. The focus in linguistics on competence rather than performance has meant that many developers of syntactic and semantic models have not traditionally evaluated their systems on a corpus of observed data. Those who have looked at data, have typically referred to a few token examples and have not evaluated systematically on an entire corpus. Still more rare is evaluation on an independent corpus, a corpus not used to derive or modify the theory or model. There is no doubt that a system can eventually be made to handle any finite number of evaluation sentences. Having a test suite of phenomena is essential for evaluating and comparing competing theories. More important for an application, however, is a test on an independent set of sentences that represent phenomena the system is likely to encounter. This ensures that developers have handled the phenomena observed in the training set in a manner that will generalize, and it properly (for systems rather than theories) focuses the evaluation of various phenomena in proportion to their likelihood of occurrence. That is, though from a theoretical perspective it may be important to cover certain phenomena, in an application, the coverage of those phenomena must be weighed against the costs (how much larger or slower is the resulting system) and benefits (how frequently do the phenomena occur).

Black Box versus Glass Box Evaluation

Evaluating components of a system is important in system development, though not necessarily useful for comparing various systems, unless the systems evaluated are very similar, which is not often the case. Since the motivation for evaluating components of a system is for internal testing, there is less need to reach wide-spread agreement in the community on the measurement methodology. System-internal measures can be used to evaluate component technologies as a function of their design parameters; for example, recognition accuracy can be tested as a function of syntactic and phonological perplexity, and parser performance can be measured as a function of the accuracy of the word input. In addition, these measures are useful in assessing the amount of progress being made, and how changes in various components affect each other.

A useful means of evaluating system performance is the time to complete a task successfully. This measure cannot be used to compare systems unless they are aimed at completing the same task. It is, however, useful in assessing the system in comparison to problem solving without the spoken language system in question. For example, if the alternative to a database query spoken language system is the analysis of huge stacks of paperwork, the simple measure of time-to-complete-task can be important in showing the efficiency gains of such a system.

Time-to-complete-task, however, is a difficult measure to use in evaluating a decision-support system because (1) individual differences in cognitive skill in the potential user population will be large in relation to the system-related differences under test, and (2) the puzzle-solving nature of the task may complicate procedures that reuse subjects as their own controls. Therefore, care should be taken in the design of such measures. For example, it is clear that when variability across subjects is large, it is important to evaluate on a large pool of users, or to use a within-subject design. The latter is possible if equivalent forms of certain tasks can be developed. In this case, each subject could perform one form of the task using the spoken language system and another form using an alternative (such as examining stacks of papers, or using typed rather than spoken input, or using a database query language rather than natural language).

Quantitative versus Qualitative Evaluation

Qualitative evaluation (for example, do users seem to like the system) can be encouraging, rewarding and can even sell systems. But more convincing to those who cannot observe the system themselves are quantitative automated measures. Automation of the measures is important because we want to avoid any possibility of nudging the data wittingly or unwittingly, and of errors arising from fatigue and inattention. Further, if the process is automated, we can observe far more data than otherwise possible, which is important in language, where the units occur infrequently and where the variation across subjects is large. For these measures to be meaningful, they should be standardized insofar as pos-

sible, and they should be reproducible. These are the goals of the DARPA-NIST protocols for evaluation of spoken language systems. These constraints form a real challenge to the community in defining meaningful performance measures.

Limiting the Domain

Spoken language systems for the near future will not handle all of English, but, rather, will be limited to a domain-specific sub-language. Accurate modeling of the sub-language will depend on analysis of domain-specific data. Since no spoken language systems currently have a wide range of users, and since variability across users is expected to be large, we are simulating applications in which a large population of potential users can be sampled.

The domain used for the standard evaluation is ATIS using the on-line Official Airline Guide (OAG), which we have put into a relational format. This application has many advantages for an initial system, including the following:

- It takes advantage of an existing public domain real database, the Official Airline Guide, used by hundreds of thousands of people.
- It is a rich and interesting domain, including data on schedules and fares, hotels and car rentals, ground transportation, local information, airport statistics, trip and travel packages, and on-time rates.
- A wide pool of users are familiar with the domain and can understand and appreciate problem solving in the domain (this is crucial both for initial data collection for development and for demonstrating the advantages of a new technology to potential future users in a wide variety of domains).
- The domain can be easily scaled with the technology, which is important for rapid prototyping and for taking advantage of advances in capabilities.
- The domain includes a good deal that can be ported to other domains, such as generic database query and interactive problem solving.

Related to the issue of limiting the domain is the issue of limiting the vocabulary. In the past, for speech recognition, we have used a fixed vocabulary. For spontaneous speech, however, as opposed to read speech, how does one specify the vocabulary? Initially, we have not fixed the vocabulary, and merely observed the lexical items that occur. However, it is an impossible task to fully account for every possible word that might occur, and it is a very large task to derive methods to detect new words. It is also a very large task to properly handle these new words, and one that probably will involve interactive systems that do not meet the requirements of our current common evaluation methods. However, there is evidence that people can accomplish tasks using a quite restricted vocabulary. Therefore, it may be possible to provide some training of subjects, and some tools in the data collection methods so that a fixed vocabulary can be specified and feedback can automatically be given to subjects when extra-lexical material occurs. This would meet the needs of spontaneous speech, of common evaluation and of a fixed vocabulary (where one could choose to include or exclude the occurring extra-lexical items in the evaluation).

Collecting Data for Evaluation

In order to collect the data we need for evaluating spoken language systems, we have developed a pnambic system (named after the line in the Wizard of Oz: "pay no attention to the man behind the curtain"). In this system a subject is led to believe that the interaction is taking place with a computer, when in fact the queries are handled by a transcriber wizard (who transcribes the speech and sends it to the subject's screen) and a database wizard who is supplied with a tool for rapid access to the online database in order to respond to the queries. The wizard is not allowed to perform complex tasks. The wizard may only retrieve data from the database or send one of a small number of other responses, such as "your query requires reasoning beyond the capabilities of the system." In general, the guidelines for the wizard are to handle requests that the wizard understands and the database can answer. The data must be analyzed afterwards to assess whether the wizard did the right thing.

The subjects in the data collection are asked to solve one of several air travel planning scenarios. The goal of the scenarios is to inspire the subjects with realistic problems and to help them focus on problem solving. A sample scenario is:

> Plan a business trip to 4 different cities (of your choice), using public ground transportation to and from the airports. Save time and money where you can. The client is an airplane buff and enjoys flying on different kinds of aircraft.

Further details on the data collection mechanism is provided in [2] in this volume.

Transcription Conventions

The session transcriptions, i.e., the sentences displayed to the subject, represent the subject's speech in a natural English text style. Errors or dysfluencies (such as false starts) that the subject corrects will not appear in the transcription. Grammatical errors that the subject does not correct (such as number disagreement) will appear in the transcription as spoken by the subject. The transcription wizard will follow general English principles, such as those described in *The Chicago Manual of Style* (13th Edition, 1982). The tremendous interactive pressure on the transcription wizard will inevitably lead

to transcription errors, so these conventions serve as a guide.

This initial transcription will then be verified and cleaned up as required. The result can be used as conventional input to text-based natural language understanding systems. It will represent what the subject "meant to say", in that it will not include dysfluencies corrected by the subject. However, it may contain ungrammatical input.

In order to evaluate the differences between previously collected read-speech corpera and the spontaneous-speech corpus, subjects will read the transcriptions of their sessions. The text used to prompt this reading will be derived from the natural language transcription while listening to the spoken input. It will obey standard textual transcriptions to look natural to the user, except where this might affect the utterance. For example, for the fare restriction code "VU/1" the prompt may appear as "V U slash one" or as "V U one", depending on what the subject said.

Finally, the above transcription needs to be further modified to take into account various speech phenomena, according to conventions for their representation. For example, obviously mispronounced words that are nevertheless intelligible will be marked with asterisks, words verbally deleted by the subject will be enclosed in angle brackets, words interrupted will end in a hyphen, some non-speech acoustic events will be noted in square brackets, pauses will be be marked with a period approximately corresponding to each elapsed second, commas will be used for less salient boundaries, an exclamation mark before a word or syllable indicates emphatic stress, and unusual vowel lengthening will be indicated by a colon immediately after the lengthened sound. Some of the indications will be useful for speech recognition systems, but not all of them will be included in the reference strings for evaluating the speech recognition output.

The various transcriptions are illustrated in the examples below, with the agreed upon file extensions in parentheses, where applicable:

- SESSION TRANSCRIPTION:
  Show me a generic description of a 757.
- NL TEXT INPUT (.nli):
  Show me a general description of a 757.
- PROMPTING TEXT (.ptx):
  Show me a general description of a seven fifty seven.
- SPEECH DETAIL (.sro):
  <list> show me: a general description, of a seven fifty seven
- SPEECH REFERENCE (.snr):
  SHOW ME A GENERAL DESCRIPTION OF A SEVEN FIFTY SEVEN

Data Classification

Once collected and processed, the data will have to be classified. Ambiguous queries will be excluded from the evaluation set only if it is impossible for a person to tell without context what the preferred reading is. Another issue is minor syntactic or semantic ill-formedness. Our guideline here is that if the query is interpretable, it will be accepted, unless it is so ill-formed that it is clear that it is not intended to be normal conversational English. All presuppositions about the number of answers (either existence or uniqueness) will be ignored, and these are the only types of presupposition failures noted to date. Any other types of presupposition failure that make the query truly unanswerable will no doubt also have made it impossible for the wizard to generate a database query, and will be ruled out on those grounds. Queries that are formed of more than one sentence will not automatically be ruled out. The examples observed so far are clearly interpretable as expressing multiple constraints that can be combined into a single query.

Evaluatable queries will be identified by exception, i.e., those that are none of the following:

1. context dependent,
2. vague, ambiguous, disambiguated only by context, or otherwise failing to yield a single canonical database answer,
3. grossly ill-formed,
4. other unanswerable queries (i.e., those r.. given a database by the wizard),
5. queries from a noncooperative subject.

Canonical Answers and Scoring

Canonical answers will, in general, be the corrected version of the answer returned under the wizard's control. These will have to be cleaned up in the case that the wizard makes an error, or if the answer given by the wizard was the (cooperative) context-dependent answer, which may differ from a context-independent answer, if it exists. In the context of a database query system, the wizard is instructed to interpret queries broadly as database requests. Thus, we believe that "yes/no" questions will be in general interpreted as a request for a list, rather than the word "yes" or "no", as in "Are there any morning flights to Denver?" Other conventions involve treatment of strings for comparison purposes and case-sensitivity, the appearance of extra columns in tabular answers, and the inclusion of identifying fields (see [1] for details).

Scoring is accomplished using standardized software, and conventions for inputs and outputs. Comparing scalar answers simply means comparing values. Table answers are more interesting, since in general the order of the columns is irrelevant to correctness. For single-element answers, a scalar answer and a table containing a single element are judged equivalent, for both specifications and answers. For our first experiment with the new protocols, sites were only required to report results on the natural language component. The transcriptions

were released a few days before the results were to be reported. One site, CMU, reported results on speech inputs. See [1] for further details on scoring.

Conclusions

The process of coming to agreement on conventions for evaluation of spoken language systems, and implementing such procedures has been a larger task than most of us anticipated. We are still learning, and sometimes it has been painful. However, the rewards of an automatic, common mechanism for system evaluation is worth the effort, and we believe the spoken language program will benefit enormously from this effort. There still is a good deal more work to do as we find ways to meet the constraints of evaluation in a way that makes sense for the development of spoken language systems.

Acknowledgements

This article is based on a perusing of the voluminous email and phone discussions involving numerous people from various sites, including BBN, CMU, MIT, MIT-LL, NIST, SRI, TI, and Unisys. The author gratefully acknowledges the important roles played by individuals from each of these sites. The program described is funded by DARPA, the particular contract that funded the writing of this paper is through DARPA under Office of Naval Research contract N00014-90-C-0085.

References

[1] L. Bates and S. Boisen, "Developing an Evaluation Methodology for Spoken Language Systems," this volume.

[2] C. Hemphill, J. Godfrey, and G. Doddington, "The ATIS Spoken Language Systems Pilot Corpus," this volume.

[3] L. Hirschman, D. Dahl, D. McKay, L. Norton, and M. Linebarger, "Beyond Class A: A proposal for Automatic Evaluation of Discourse," this volume.

[4] D. Pallett and W. Fisher, "Performance Results Reported to NIST," this volume.

[5] D. Pallett, chair, "ATIS Site Reports and General Discussion," Session 5, this volume.

[6] P. J. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *Proc. ICASSP*, 1988. Database available on CD-ROM.

[7] B. Sondheim, "Plans for a Task-Oriented Evaluation of Natural Language Understanding Systems." *Proc. of the DARPA Speech and Natural Language Workshop*, Feb. 1989.

Proc. ICSLP, Kobe, Japan, 1990

Spoken Language System Integration and Development

**Patti Price, Victor Abrash, Doug Appelt, John Bear, Jared Bernstein, Bridget Bly,
John Butzberger, Michael Cohen, Eric Jackson, Robert Moore,
Doug Moran, Hy Murveit, and Mitchel Weintraub**

SRI International, Menlo Park, California 94025 USA

Abstract

SRI is developing a spoken language system (SLS) that should permit natural and efficient communication with an air travel information system. SLS development at SRI divides roughly into three areas: speech recognition, natural language processing, and human interface design. The paper presents an overview of SRI's development effort and an analysis of selected technical challenges in subparts of this effort, including the choice of initial domains for such technology, the architecture for the integration of the two technologies, the attributes of goal-directed spontaneous speech, and the evaluation of spoken language systems.

1.0 Introduction

Combining speech recognition and natural language understanding will vastly increase the number and range of potential applications for both technologies. Speech recognition without natural language results in a transcription of the words spoken; adding an interpretation of what those words mean opens a vast range of possibilities in human-machine interaction. Natural language technology without speech recognition requires typing skills and makes unnecessary demands on the eyes, the hands, and the brain. Freeing the eyes, hands, and brain of the user from the keyboard will allow for more efficiency, better use of visual displays and mouse interactions, interactive problem solving during hands-busy tasks, and flexible telephone applications. By using spoken natural language, the user can focus more on the problem to be solved and less on how to formulate it adequately for the computer.

A further motivation for the integration of speech recognition and natural language understanding is the belief that each technology could be improved by taking advantage of the other. Not every word can follow every other word. This is true in any language. Grammars are expressions of conditions on possible word sequences. Constraining the possible, or likely, sequences of words has had a major impact on large-vocabulary speech recognition because it effectively reduces the work done by the recognizer and eliminates many otherwise possible sources of confusion. Taking advantage of the grammatical constraints of a language could be important in improving speech recognition performance. With the exception of small domain-dependent grammars, such constraint to date typically comes from models of the statistical properties of word sequences. Such grammars have difficulty expressing constraints that are based on grammatical relations that may span an arbitrary number of words. Just as natural language constraints could improve speech recognition, information from speech could improve natural language understanding: Speech includes much information that is not indicated in the text, such as lexical, phrasal and contrastive stress, and prosodic groupings of words. Such information can aid lexical decisions (e.g., is the word "OBject" or "obJECT") as well as syntactic and semantic decisions.

The attempt to go beyond speech transcription and to go beyond text understanding by moving toward spoken language understanding opens an exciting new array of possibilities for human-machine interaction. It also opens a new array of issues not previously faced. The issues discussed in this paper include:

- The choice of initial domains for such technology
- The architecture for the integration of the two technologies
- The attributes of goal-directed spontaneous speech
- Evaluation of spoken language systems.

2.0 Domains

Spoken language understanding is a technology in its infancy. The first systems will be extremely limited, and we have little experience in the human factors issues of integrating the technology into an application. Spoken language understanding is an exciting area for human-machine interaction because people are used to solving problems interactively by voice. For this same reason, however, adding spoken language understanding to an interface may lead the user to believe the system has reasoning and understanding capabilities beyond current achievements.

Designing the human interface for inserting a new technology in an application is difficult, since we have no existing systems to observe. A promising technique for gaining the required data on human-machine interactions is the use of simulations of applications. Since variability across users in speech and language is quite large, initial systems should focus on applications in which a large population of potential users can be sampled. The data thus obtained can be used to develop initial systems and to develop methods for obtaining more such data efficiently for future systems.

The domain SRI has chosen for its first spoken-language, interactive, problem solving system is air travel planning. This domain has several important advantages as a first area:

- It takes advantage of an existing public domain real database, the *Official Airline Guide*, used by hundreds of thousands of people in the United States.
- It is a rich and interesting domain, including data on schedules and fares, hotels and car rentals, ground transportation, local information, airport statistics, trip and travel packages, on-time rates, and so on.
- A wide pool of users are familiar with the domain and can understand and appreciate problem solving in the domain. (This is crucial both for initial data collection for development and for demonstrating the advantages of a new technology to potential future users in a wide variety of domains.)
- The domain can be easily scaled with the technology, which is important for rapid prototyping and for taking advantage of advances in capabilities.
- The domain includes a significant amount that can be ported to other domains, such as generic database query and interactive problem solving.

3.0 Integration Architecture

3.1 Previous Approaches

A speech recognition component might communicate in several different ways with a natural language understanding component. Perhaps the most straightforward approach is a serial connection. In this scheme, the speech is input to the recognition system which, on the basis of the speech alone, outputs its best hypothesis to the natural language understanding system, which computes a meaning on the basis of text alone. There is no feedback in this scheme: the speech component does not have access to syntax and semantics in hypothesizing words, and the natural language component does not have access to, for example, the prosody of the speech for understanding contrastive stress. This approach has the advantage of being simple and of putting no additional effort into either of the two component technologies. It also has the advantage of requiring minimal communication between two culturally distinct groups: the engineers that dominate the speech recognition community and the artificial intelligence community that dominates natural language understanding.

Serial integration is, however, suboptimal because it does not take advantage of all the information available. A sentence that is misrecognized may have little hope of receiving a proper interpretation. We know that humans use a good deal of knowledge about syntax and semantics in interpreting what another person has said. A spoken language system should be able to take advantage of this information as well. Modifications to the strict serial architecture include sending a large lattice of words from the speech recognition component or a sequence of sentence hypotheses. This allows the syntax or semantics to explore more than just the best speech hypothesis. Sending a large lattice can reduce the error rate, provided the correct set of words is somewhere in the lattice or sentence list. Architectures of this type have been explored (Schwartz & Chow 1989; Paul 1989). However, a tighter integration should improve performance by allowing more communication among the components earlier in the process.

3.2 SRI's Frame-level Integration

More communication between the speech and the understanding components involve more complex architectures, but should improve both the speed and the accuracy of the spoken language system. SRI is investigating a unique frame-level integration (information between the two components is exchanged every 10 msec) that allows a computationally efficient use of natural language constraints in the speech recognition search. This system architecture allows for independent development yet integrated application of constraints from phonetics through semantics.

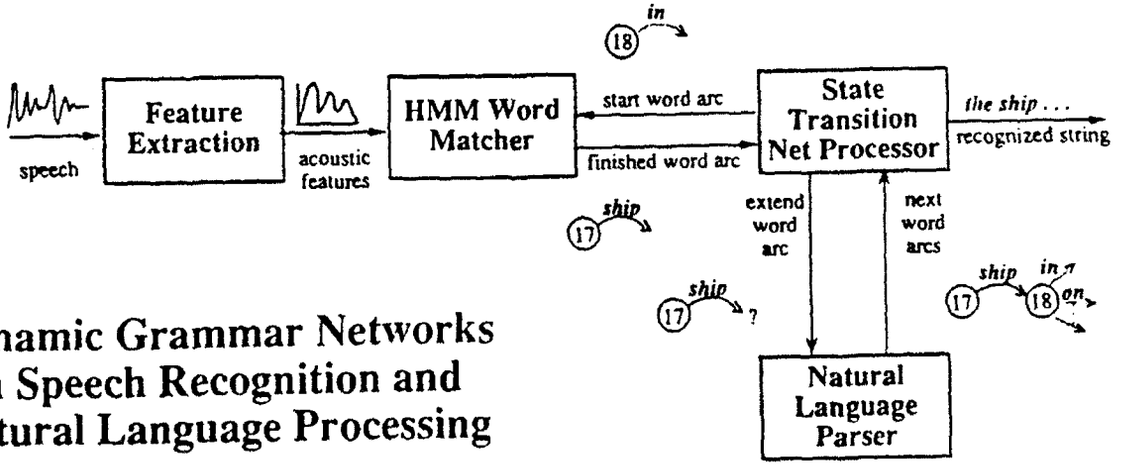
SRI's approach, called *dynamic grammar network* (DGN) generation (Murveit & Moore 1990), represents natural language knowledge in a state transition network, similar to finite-state language models used elsewhere for speech recognition systems. A straightforward implementation of this approach is not feasible, however, because typical NL systems would generate unmanageably large or infinite networks. Therefore, the network is generated on the fly, and only the portions of the network within a pruned search are expanded. Thus, the state-transition network generated for a particular spoken sentence will be relatively small, and different from that generated for a different utterance.

The approach is described graphically in Figure 1. The system runs as if it were a standard speech recognition system based on a hidden Markov model (HMM) using a language model based on a state-transition network. When the system is started up, the state-transition network contains an initial state, a list of the words that can leave that state (predictions), and markers indicating that the states that would be reached from these initial predictions are blocked--not yet included in the state transition network. The speech recognition (SR) system begins by searching for the words in the initial state's prediction list using a standard beam search. When a state is reached that is not in the network, the SR system calls the natural language processing (NLP) system which runs the parser, and creates the needed state. The SR system can then continue until it blocks again. The process of accepting the completion of a word from a state in the network and generating a new state is called a shift, as it corresponds to a shift in a *shift-reduce* natural language parser (Aho & Ullman 1979).

The shift process continues until the entire signal is exhausted. Words ending at the end of the signal are checked to see if they reach a final state--a state such that the hypotheses reaching that state are acceptable as complete utterances--and the most probable final-state hypothesis is chosen as the recognized sentence.

This approach allows a tight coupling of SR and NLP algorithms and has the following advantages:

- It brings all knowledge to bear as soon as possible so that extra work need not be done (for instance, the recognizer will not pursue hypotheses that can be ruled out by NLP and vice versa). In contrast to an equivalent system based on word lattices, a dynamic-grammar network system would not search portions of the signal that correspond to word-lattice entries that are unlikely according to previous acoustics or natural language.
- It allows for interactions between speech and NLP. For instance, an acoustic recognition model can be altered if the NLP system judges that the word should be emphasized given its syntactic or semantic position.



Dynamic Grammar Networks in Speech Recognition and Natural Language Processing

- In addition, this approach has the important advantage that, from the perspective of the recognition system, finite-state language constraints are used. Thus, all of the experience the speech recognition community has developed for dealing with finite-state-based speech recognition systems still applies to this system. For instance, a standard beam-search pruning technique is used in this system (Lowerre 1976).

4.0 Goal-Directed Speech

When a person is dictating to a system the goal is to communicate the words; the speaker is more likely to enunciate carefully and to focus on how the words are produced. When, however, a person is involved in interactive problem solving, the focus is not, or should not be, on the speech itself, but on the problem to be solved. This means that the speech is likely to be less careful and more casual. In particular, this means that there may be more variability in pronunciation, and that segments and syllables may be more likely to be reduced or deleted. It also means that more instances of "non-standard" grammatical forms will occur.

4.1 Phonological Variation

SRI has partially addressed the issue of phonological variation by incorporating detailed, statistically trained models of possible pronunciations for words (Cohen 1989, Cohen et al. 1990). The rules for pronunciation variations are created once for English and then can be applied to automatically generate a network of possible pronunciations for any new word. The likelihoods of the variants can also be automatically estimated on the basis of observations of the occurrences of similar instances in training data that need not contain the new words.

4.2 Grammatical Variation

The common production of non-standard grammatical forms brings into focus the trade-off between complete understanding of a given utterance and reliance on alternative techniques for interpretation. Even within a restricted domain, full understanding of any utterance, is difficult to accomplish. Language is productive, so new constructions appear frequently. Further, people often get distracted or change their minds in mid-sentence, which can result in wide deviations from "standard" language structure. Therefore, it seems useful to allow some flexibility in what the grammar will allow. However, accommodating more constructions typically requires more computation (and longer waiting time for the user), and also will provide less constraint (and thus make greater demands of accuracy on the recognition component). One solution to this problem is to bring more knowledge sources to bear, such as dialogue or plan models. However, a new domain has little data available on which to base a plan model, and poor models can perform worse than no model at all. At SRI we are exploring various combinations of tight and flexible grammars, trying to obtain the advantages of both. For the time being, SRI is pursuing the idea of cascading an analytical, linguistically-based grammar with a template-filler grammar so that the template filler can analyze those sentences that the analytical system cannot handle.

4.3 Template Grammar

In our initial work in this area, we have constructed a template-based grammar based on an analysis of frequently occurring patterns in the air travel planning domain.

We created templates corresponding to several common types of information that can be produced by the system (for example, schedules of flights, fares, seat availability, etc.) Templates are triggered based on the existence of keywords within a sentence, and multiple templates can be triggered for the same sentence. Templates contain slots such as the origin and destination of a flight in question. The slots are filled in from phrases following slot-keywords. Thus, for example, in the sentence "What flights leave San Francisco for Boston on Sunday?" the word flights will be a keyword triggering the "Flights" template, "leave" will cause the next phrase (if it is a city or airport) to be placed in the from-slot, "for" will cause the next phrase (if it is a city or airport) to be placed in the to-slot, and on (if the next phrase is a time) will cause the time slot for the flights question to be filled.

Template hypotheses are scored according to the percentage of content words used in filling the slots of the template. The template with the highest score is selected for interpretation. However, this grammar has a "cut-off" parameter for template scores that can be set to trade off wrong answers with no answers. That is, when the system is unsure, it can either guess, or admit that it doesn't know. Different applications would require different settings of this parameter. Our initial results with this system are very encouraging. On a fair test (testing on data not used in development) using DARPA standards for evaluation, we recently obtained the results shown in Table 1 for various settings of the cut-off parameter.

TABLE 1 PARSING PERFORMANCE AS A FUNCTION OF CUT-OFF

Cut-off	Right	Wrong	No Answer
0.0	55	13	22
0.833	42	4	44
1.0	37	2	51

These are very preliminary results, and much work remains to be done to combine the two grammars.

5.0 Evaluation

Progress can be measured and encouraged via standards for comparison and evaluation. Although qualitative assessments can be useful in initial stages, quantifiable measures of systems under the same conditions are essential for comparing results and assessing claims. Numbers are meaningless unless it is clear where they come from. The evaluation of any technology is greatly enhanced in usefulness if accompanied by documented standards for assessment. There has been a growing appreciation in the speech recognition community of the importance of standards for reporting performance. The availability of standard databases and protocols for evaluation has been an important component in progress in the field and in the sharing of new ideas. Progress toward evaluating spoken language systems, like the technology itself, is beginning to emerge. The following issues have been important in coming to agreement on standards for evaluation.

5.1 Independent Training and Test Sets

The importance of independent training/development data and testing data has been acknowledged in speech recognition evaluation for some time. The idea is less prominent in natural language understanding because, from a theoretical perspective, it may be important to work on a certain class of phenomena. In an application, however, the coverage of a certain class of phenomena must be weighed against the costs (how much larger or slower is the resulting system) and benefits (how frequently do the phenomena occur). The only fair test of coverage in this sense is a test on a sample of data similar to that to be used in the application, but not seen during development.

5.2 Black Box versus Glass Box Evaluation

Evaluating components of a system is important in system development, although not necessarily useful for comparing various systems, unless the systems evaluated are very similar, which is not often the case. Since the motivation for evaluating components of a system is for internal testing, there is less need to reach wide-spread agreement in the community on the measurement methodology. System-internal measures can be used to evaluate component technologies as a function of their design parameters; for example, recognition accuracy can be tested as a function of syntactic and phonological perplexity, and parser performance can be measured as a function of the accuracy of the word input. In addition, these measures are useful in assessing the amount of progress being made, and how changes in various components affect each other.

5.3 Quantitative versus Qualitative Evaluation

Qualitative evaluation (for example, do users seem to like the system) can be encouraging, but more convincing to those who cannot observe the system themselves are quantitative automated measures. Automation of the measures is important because we want to avoid any possibility of nudging the data wittingly or unwittingly, and of errors arising from fatigue and inattention. Further, if the process is automated, we can observe far more data than otherwise possible, which is important in research on language, where many units occur infrequently and where the variation across subjects can be large. For these measures to be meaningful, they should be standardized insofar as possible, and they should be reproducible.

5.4 Collecting Data for Evaluation

In order to collect the data we need for evaluating spoken language systems, we have developed a *pnambic* system (named after the line in the *Wizard of Oz*: "pay no attention to the man behind the curtain"). In this system a subject is led to believe that the interaction is taking place with a computer, when in fact the queries are handled by a transcriber wizard (who transcribes the speech and sends it to the subject's screen) and a database wizard who is supplied with a tool for rapid access to the on-line database in order to respond to the queries. The wizard is not allowed to perform complex tasks. The wizard may only retrieve data from the database or send one of a small number of other responses, such as "your query requires reasoning beyond the capabilities of the system." In general, the guidelines for the wizard are to handle requests that the wizard understands and the database can answer. The data must be analyzed afterwards to assess whether or not the wizard did the right thing.

5.5 Transcription Conventions

The session transcriptions, i.e., the sentences displayed to the subject, represent the subject's speech in a natural English text style. In order to perform automatic evaluation, we must agree on conventions for representing what the subject said, and we must implement procedures to ensure that these conventions are consistently used.

5.6 Canonical Answers and Scoring.

Canonical answers are, in general, the answer returned under the wizard's control. These answers will have to be cleaned up if the wizard makes an error, or if the answer given by the wizard was the (cooperative) context-dependent answer, which may differ from a context-independent answer, if it exists. Scoring is accomplished using standardized software, and conventions for inputs and outputs.

The process of coming to agreement on conventions for evaluation of spoken language systems, and implementing such procedures is difficult and time-consuming. However, the rewards of an automatic, common mechanism for system evaluation is worth the effort, and we believe that spoken language system development will benefit enormously from this effort.

6.0 Summary

In sum, workstations equipped with spoken language systems have the potential to increase user efficiency in interactive problem-solving. Natural language input allows the user to formulate more complex questions and commands more efficiently and more naturally. Spoken natural language can increase user efficiency, can reduce cognitive load, and can provide an alternate input modality to improve system robustness. SRI's research suggests that successful development of SLS technology requires an appreciation of the new challenges associated with acceptance of user input that cannot be defined beforehand. Furthermore, system integration design decisions can affect how well the system can deal with these new input forms.

References

A. Aho and J. Ullman (1979) *Principles of Compiler Design*. Addison-Wesley, Reading Mass.

M. Cohen (1989) "Phonological Structures for Speech Recognition," PhD Thesis, Computer Science Dept., University of California, Berkeley.

M. Cohen, H. Murveit, J. Bernstein, P. Price and M. Weintraub (1990) "The DECIPHER Speech Recognition System," *Proc. IEEE ICASSP-90*.

B. Lowerre (1976) *The Harpy Speech Recognition System*. PhD Thesis, Computer Science Dept., Carnegie Mellon U.

H. Murveit and R. Moore (1990) "Integrating Natural Language Constraints into HMM-based Speech Recognition," *Proc. IEEE ICASSP-90*.

D. Paul (1989) "A CSR/NLP Interface Specification," *Proc. of the DARPA Speech and Natural Language Workshop*, Oct. 1989.

R. Schwartz & Y-R Chow (1989) "The optimal N-Best Algorithm: An Efficient Procedure for Finding the Top N Sentence Hypotheses," *Proc. of the DARPA Speech and Natural Language Workshop*, Oct. 1989.

We gratefully acknowledge support from SRI internal funding, DARPA and NSF. Support from DARPA is through the Office of Naval Research contract N00014-90-C-0085. This material is based upon work supported by the National Science Foundation under Grant No. IRI-87204403. The government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

SUBJECT-BASED EVALUATION MEASURES FOR INTERACTIVE SPOKEN LANGUAGE SYSTEMS

Patti Price,[1] Lynette Hirschman,[2] Elizabeth Shriberg,[3] Elizabeth Wade[4]

[1]SRI International, 333 Ravenswood Ave., EJ 133, Menlo Park, CA 94306.
[2] MIT Laboratory for Computer Science, Cambridge, MA 02139
[3]University of California at Berkeley, Department of Psychology, Berkeley, CA94720
[4]Stanford University, Department of Psychology, Stanford, CA 94305

ABSTRACT

The DARPA Spoken Language effort has profited greatly from its emphasis on tasks and common evaluation metrics. Common, standardized evaluation procedures have helped the community to focus research effort, to measure progress, and to encourage communication among participating sites. The task and the evaluation metrics, however, must be consistent with the goals of the Spoken Language program, namely interactive problem solving. Our evaluation methods have evolved with the technology, moving from evaluation of read speech from a fixed corpus through evaluation of isolated canned sentences to evaluation of spontaneous speech in context in a canned corpus. A key component missed in current evaluations is the role of subject interaction with the system. Because of the great variability across subjects, however, it is necessary to use either a large number of subjects or a within-subject design. This paper proposes a within-subject design comparing the results of a software-sharing exercise carried out jointly by MIT and SRI.

1. INTRODUCTION

The use of a common task and a common set of evaluation metrics has been a cornerstone of DARPA-funded research in speech and spoken language systems. This approach allows researchers to evaluate and compare alternative techniques and to learn from each other's successes and failures. The choice of metrics for evaluation is a crucial component of the research program, since there will be strong pressure to make improvements with respect to the metric used. Therefore, we must select metrics carefully if they are to be relevant both to our research goals and to transition of the technology from the laboratory into applications.

The program goal of the Spoken Language Systems (SLS) effort is to support human-computer interactive problem solving. The DARPA SLS community has made significant progress toward this goal, and the development of appropriate evaluation metrics has played a key role in this effort. We have moved from evaluation of closed vocabulary, read speech (resource management) for speech recognition evaluation to open vocabulary for spontaneous speech (ATIS). In June 1990, the first SLS dry run evaluated only transcribed spoken input for sentences that could be interpreted independent of context. At the DARPA workshop in February 1991, researchers reported on speech recognition, spoken language understanding, and natural language understanding results for context-independent sentences and also for pairs of context-setting + context-dependent sentences. At the present workshop, we witness another major step: we are evaluating systems on speech, spoken language and natural language for all evaluable utterances within entire dialogues, requiring that systems handle each sentence in its dialogue context, with no externally supplied context classification information.

2. EVALUATION METHODOLOGY: WHERE ARE WE?

The current measures have been and will continue to be important in measuring progress, but they do not assess the interactive component of the system, a component that will play a critical role in future systems deployed in real tasks. Indeed, some current metrics may penalize systems that attempt to be co-operative (for example, use of the weighted error, see below, and the maximal answer constraints). We propose a complementary evaluation paradigm that makes possible the evaluation of interactive systems. In this section we outline the current state of evaluation methodology and point out some shortcomings.

The current evaluation procedure is fully automated, using a canned corpus as input and a set of canonical database tuples as output reference answers. The evaluation measures the recognition and understanding components of a spoken language system, based on the number of correctly answered, incorrectly answered, and unanswered queries. These are then incorporated into a single number to produce a weighted error: percent "No-Answer" plus twice the percent "Incorrect" (this formulation is equivalent to 1 minus the "Score", where the "Score" is the percent "Correct" minus the percent "Incorrect)."

To date, evaluation has focused on only half of the user-to-system-to-user loop: these measures evaluate how well the system understands what the user says, but they ignore how felicitously the system answers the user. The use of a canned corpus effectively precludes any attempt by the system to direct the course of the conversation, including eliciting missing information from the subject, carrying on a clarification dialogue (for ambiguous or misunderstood input) or providing partial answers. For example, suppose the user says "I want to go to Dallas next Sunday". There are a number of ways for the system to respond:

1. Ask for a place of departure:

   SYSTEM: "Where are you leaving from?"

2. Inform the user that the information is incomplete:

   SYSTEM: "Please give both a place of departure and a destination."

3. Show all flights arriving in Dallas from anywhere:

   SYSTEM: "Here are the flights arriving in Dallas from anywhere."

4. Refuse to answer the question, with some diagnostic message:

   SYSTEM: "I'm sorry, there are too many flights to show; please give me more information."

5. Refuse to answer the question:

   SYSTEM: "Please rephrase your question."

Obviously some of these responses seem more helpful than others (although it is not immediately clear that there is a "best" response). In any case, what the user says next depends on which strategy the system adopts. For example, if the first strategy is chosen, where the system asks the user to specify a departure city, the user might well respond "Boston." This makes sense as an answer to System Response 1, but is not easily interpreted if the system uses System Response 2: "Please give both a place of departure and a destination." Maintaining coherence in dialogue depends on what the system says to the user during data collection. If we continue to use only canned dialogue in evaluation, then we can choose among ways to evaluate systems:

1. Canonical Response: All systems must provide the identical response to each situation;

2. Dialogue Resynchronization: Each system must be able to process the data collection system's response, and resynchronize its state based on what the user received as a response during data collection;

3. Discarding Interactive Dialogue: We throw out of the common evaluation all interactive (particularly mixed initiative) dialogue.

Alternative 1 was rejected as stifling experimentation in building interactive systems. Alternative 2 was described in [6], but some sites have felt that it was too burdensome to implement (multiparty dialogue tracking is not an immediate requirement for our current spoken language systems). We are left with Alternative 3, disallowing interactive dialogue in evaluation. This is cause for concern, since using spoken language systems for interactive problem solving is a shared goal [9]. It also leaves open the possibility that we inadvertently direct our research toward strategies that maximize our current measures but may not translate to any measurable gains when used in real-world applications.

The push toward a single-number, automated, evaluation score has encouraged system builders to adopt strategies that maximize this score. To arrive at a single number, it is necessary to combine right answers, wrong answers, and "no answers". The current formula makes a wrong answer twice as bad as saying "I don't understand." As indicated by Appelt at the October 1991 Spoken Language Interim Meeting, these strategies may not always be the most helpful to a user interacting with the system to accomplish a task. For example, in our current evaluation framework, the answer in the following interchange could be counted as wrong, because the information would be insufficiently restricted:

USER: Show me flights from Boston to Dallas leaving tomorrow on some kind of wide-body airplane

SYSTEM: You want to see flights from Boston to Dallas leaving tomorrow, but I didn't understand the rest of the sentence; here are tomorrow's flights from Boston to Dallas: (*shows the flights but doesn't restrict answer to wide-bodies*).

Similarly, the following interchange would typically be scored correct:

USER: Show me the flights to Boston arriving before 7 a.m.

SYSTEM: Here are the flights from anywhere to Boston arriving before 7 a.m.: (*shows all flights to Boston from anywhere arriving before 7 a.m.*).

Meanwhile the following interaction counts as "unevaluable" because the system takes the initiative, providing a (possibly irreproducible) context necessary to interpret the user's next input:

USER: Show me the flights to Boston arriving before 7 a.m.

SYSTEM: Where are you departing from?

USER: Dallas

SYSTEM: Here are the flights to Boston from Dallas arriving before 7 a.m.:

To move this technology toward realistic applications, we need to focus on building useful systems. This requires that we explore trade-offs between guessing, partial answers, reporting lack of understanding, and having a clarification dialogue with the user. We also need to explore different styles of interaction between system and user. In addition, most aspects of the system interface (particularly, the form of the output) are not being assessed at all using current metrics (e.g., display of information, presence or absence of spoken output, mixed initiative strategies). We need to develop complementary evaluation techniques that allow us to make progress and measure performance on interactive systems, rather than confining ourselves to a metric that may penalize cooperativeness. Further, we need a sanity check on our measures to reassure ourselves that gains we make according to the measures will translate to gains in application areas. The time is right for this next step, now that many sites have real-time spoken language systems.

3. METHODS

We have argued that interactive systems cannot be evaluated solely on canned input; live subjects are required. However, live subjects can introduce uncontrolled variability across users which can make interpretation of results difficult. To address this concern, we propose a within-subject design, in which each subject solves a scenario using each system to be compared, and the scenario order and system order are counterbalanced. However, the within-subject design requires that each subject have access to the systems to be compared, which means that the systems under test must all be running in one place at one time (or else that subjects must be shipped to the sites where the systems reside, which introduces a significant time delay). Given the goal of deployable software, we chose to ship the software rather than the users, but this raises many infrastructure issues, such as software portability and modularity, and use of common hardware and software.

Our original plan was to test across three systems: the MIT system, the SRI system, and a hybrid SRI-speech/MIT-NL system. SRI would compare the SRI and SRI-MIT hybrid systems; MIT would compare the MIT and SRI-MIT hybrids. The first stumbling block was the need to license each system at the other site; this took some time, but was eventually resolved. The next stumbling block was use of site-specific hardware and software. The SRI system used D/A hardware that was not available at MIT. Conversely, the MIT system required a Lucid Lisp license, which was not immediately available to the SRI group. Further, research software typically does not have the documentation, support, and portability needed for rapid and efficient exchange. Eventually, the experiment was pared down to comparing the SRI system and the SRI/MIT hybrid system at SRI. These infrastructure issues have added considerable overhead to the experiment.

The SRI SLS employs the DECIPHER™ speech recognition system [4] serially connected to SRI's Template Matcher system [7,1]. The pruning threshold of the recognizer was tuned so that system response time was about 2.5 times utterance duration. This strategy had the side-effect of pruning out more hypotheses than in the comparable benchmark system, and a higher word error rate was observed as a consequence. The system accesses the relational version of the Official Airline Guide database (implemented in Prolog), formats the answer and displays it on the screen. The user interface for this system is described in [16]. This system, referred to as the SRI SLS, will be compared to the hybrid SRI/MIT SLS. The hybrid system employs the identical version of the DECIPHER recognizer, set at the same pruning threshold. All other aspects of the system differ. In the SRI/MIT hybrid system, the DECIPHER recognition output is connected to MIT's TINA [15] natural-language understanding system and then to MIT software for database access, response formatting, and display. Thus, the experiment proposed here compares SRI's natural language (NL) understanding and response generation with the same components from MIT. We made no attempt to separate the contribution of the NL components from those of the interface and display, since the point of this experiment was to debug the methodology; we simply cut the MIT system at the point of easiest separation. Below, we describe those factors that were held constant in the experiment and the measures to be used on the resulting data.

3.1. Subjects, Scenarios, Instructions

Data collection will proceed as described in Shriberg et al. 1992 [16] with the following exceptions: (1) updated versions of the SRI Template Matcher and recognizer will be used; (2) subjects will use a new data collection facility (the room is smaller and has no window but is acoustically similar to the room used previously); (3) the scenarios to be solved have unique solutions; (4) the debriefing questionnaire will be a merged version of the questions used on debriefing questionnaires at SRI and at MIT in separate experiments; and (5) each subject will solve two scenarios, one using the SRI SLS and one using the SRI/MIT hybrid SLS. Changes from our previous data collection efforts are irrelevant as all comparisons will be made within the experimental paradigm and conditions described here.

MIT designed and tested two scenarios that were selected for this experiment:

SCENARIO A. Find a flight from Philadelphia to Dallas that makes a stop in Atlanta. The flight should serve breakfast. Find out what type of aircraft is used on the flight to Dallas. Information requested: aircraft type.

SCENARIO B. Find a flight from Atlanta to Baltimore. The flight should be on a Boeing 757 and arrive around 7:00 p.m. Identify the flight (by number) and what meal is served

on the flight. Information requested: flight number, meal type.

We will counterbalance the two scenarios and the two systems by having one quarter of the subjects participate in each of four conditions:

1. Scenario A on SRI SLS, then Scenario B on SRI/MIT hybrid SLS
2. Scenario A on SRI/MIT hybrid SLS, then Scenario B on SRI SLS
3. Scenario B on SRI SLS, then Scenario A on SRI/MIT hybrid SLS and
4. Scenario B on SRI/MIT hybrid SLS, then Scenario A on SRI SLS).

A total of 12 subjects will be used, 3 in each of the above conditions. After subjects complete the two scenarios, one on each of the two systems, they will complete a debriefing questionnaire whose answers will be used in the data analysis.

3.2. Measures

In this initial experiment, we will examine several measures in an attempt to find those most appropriate for our goals. One measure for commercial applications is the number of units sold, or the number of dollars of profit. Most development efforts, however, cannot wait that long to measure success or progress. Further, to generalize to other conditions, we need to gain insight into why some systems might be better than others. We therefore chose to build on experiments described in [12] and to investigate the relations among several measures, including:

- User satisfaction. Subjects will be asked to assess their satisfaction with each system (using a scale of 1-5) with respect to the scenario solution they found, the speed of the system, their ability to get the information they wanted, the ease of learning to use the system, comparison with looking up information in a book, etc. There will also be some open-ended questions in the debriefing questionnaire to allow subjects to provide feedback in areas we may not have considered.
- Correctness of answer. Was the answer retrieved from the database correct? This measure involves examination of the response and assessment of correctness. As with the annotation procedures [10], some subjective judgment is involved, but these decisions can be made fairly reliably (see [12] for a discussion on interevaluator agreement using log file evaluation). A system with a higher percentage of correct answers may be viewed as "better." However, other factors may well be involved that correctness does not measure. A correlation of correctness with user satisfaction will be a stronger indication of the usefulness of this measure. Lack of correlation might reveal an interaction with other important factors.
- Time to complete task, as measured from the first push-to-talk until the user's last system action. Once task and subject are controlled, as in the current design, making this measurement becomes meaningful. A system which results in faster completion times may be preferred, although it is again important to assess the correlation of time to completion with user satisfaction.
- User waiting time, as measured between the end of the first query and the appearance of the response. Faster recognition has been shown to be more satisfying [16] and may correlate with overall user satisfaction.
- User response time, as measured between the appearance of the previous response and the push-to-talk for the next answer. This time may include the time the user needs to formulate a question suitable for the system to answer as well as the time it takes the user to assimilate the material displayed on the screen. In any case, user response time as defined here is distinct from waiting time, and is a readily measurable component of time to completion.
- Recognition word error rate for each scenario. Presumably higher accuracy will result in more user satisfaction, and these measures will also allow us to make comparison with benchmark systems operating at different error rates.
- Frequency and type of diagnostic error messages. Systems will typically display some kind of message when it has failed to understand the subject. These can be automatically logged and tabulated.

4. SUMMARY AND DISCUSSION

As pointed out by LTC Mettala in his remarks at this meeting, we need to know more than the results of our current benchmark evaluations. We need to know how changes in these benchmarks will change the suitability of a given technology for a given application. We need to know how our benchmarks correlate with user satisfaction and user efficiency. In a sense, we need to evaluate our evaluation measures.

At this writing, the MIT software has been transferred to SRI, and data collection is about to begin. We find that what began as an exercise in evaluation has become an exercise in software sharing. We do not want to deny the importance of software sharing and its role in strengthening portability. However, the difficulties involved (legal and other paperwork, acquisition of software and/or hardware, extensive interaction between the two sites) are costly enough that we believe we should also consider mechanisms that achieve our goals without requiring exchange of complete systems. Two such possibilities are described below.

Existing logfiles, including standard transcriptions, could be presented to a panel of evaluators for judgments of the appropriateness of individual answers and of the interaction as a whole. In a sense, then, the evaluators would simulate different users going through the same problem solving experience as the subject who generated the logfile. Cross-site variability of subjects used for this procedure could be somewhat controlled by specifying characteristics of these subjects (first time users, 2 hours of experience, daily computer user, etc.). This approach has several important advantages:

- It allows a much richer set of interactive strategies than our current metrics can assess, which can spur research in the direction of the stated program goals.
- It provides an opportunity to assess and improve the correlation of our current metrics with measures that are closer to the views of consumers of the technology, which should yield greater predictive power in matching a given technology to a given application.
- It provides a sanity check for our current evaluation measures, which could otherwise lead to improved scores but not necessarily to improved technology.
- It allows the same scenario-session to be experienced by more than one user, which addresses the subject-variability issue.
- It requires no exchange of software or hardware, and takes advantage of existing data structures currently required of all data collection sites, which means it is relatively inexpensive to implement.

The method however does NOT make use of a strictly within-subject design, i.e., the same subject does not interact with different systems (although the same evaluator would assess different systems). As a result, the logfile evaluation may require use of more subjects, or other techniques for addressing the issue of subject variability.

A live evaluation in which sites would bring their respective systems to a common location for assessment by a panel of evaluators could provide a means for a within-subject design. The solution of having a live test would have benefits similar to those outlined above for the logfile evaluation, but in addition subjects could assess the speed of system response, which the logfile proposal largely ignores. However, it would be more costly to transport the systems and the panel of evaluators than to ship logfiles (although most sites currently bring demonstration systems to meetings).

The logfile proposal could be modified to overcome its limited value in assessment of timing (at some additional expense) by the creation of a mechanism that would play back the logfiles using a standard display mechanism and based on the time stamps appearing in the logfiles. This would also open the possibility of having evaluators hear the speech of the subject, rather than just seeing transcriptions.

The costs involved for the use of such measures is negligible given the potential benefits. We propose these methods not as a replacement for the current measures, but rather as a complement to them and as a reality check on their function in promoting technological progress.

Acknowledgment. We gratefully acknowledge support for the work at SRI by DARPA through the Office of Naval Research Contract N00014-90-C-0085 (SRI), and Research Contract N00014-89-J-1332 (MIT). The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the government funding agencies. We also gratefully acknowledge the efforts of David Goodine of MIT and of Steven Tepper at SRI in the software transfer and installation. This research was supported by DARPA.

References

1. Appelt, D., Jackson, E., and R. Moore, "Integration of Two Complementary Approaches to Natural Language Understanding," *Proc. Fifth DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

2. Bates, M., Boisen, S., and J. Makhoul, "Developing an Evaluation Methodology for Spoken Language Systems," pp. 102-108 in *Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

3. Bly, B., P. Price, S. Tepper, E. Jackson, and V. Abrash, "Designing the Human Machine Interface in the ATIS Domain," pp. 136-140 *in Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

4. Butzberger, J., H. Murveit, M. Weintraub, P. Price, and E. Shriberg, "Modeling Spontaneous Speech Effects in Large Vocabulary Speech Applications," *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

5. Hemphill, C. T., J. J. Godfrey, and G. R. Doddington, "The ATIS Spoken Language System Pilot Corpus," pp. 96-101

in Proc. Third Darpa Speech and Language Workshop, Morgan Kaufmann, 1990.

6. Hirschman, L., D. A. Dahl, D. P. McKay, L. M. Norton, L., and M. C. Linebarger, "Beyond Class A: A Proposal for Automatic Evaluation of Discourse," pp. 109-113 *in Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

7. Jackson, E., D. Appelt, J. Bear, R. Moore, A. Podlozny, "A Template Matcher for Robust NL Interpretation," pp. 190-194 in *Proc. Fourth DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

8. Kowtko, J. C. and P. J. Price, "Data Collection and Analysis in the Air Travel Planning Domain," pp. 119-125 in *Proc. Second Darpa Speech and Language Workshop*, Morgan Kaufmann, 1989.

9. Makhoul, J., F. Jelinek, L. Rabiner, C. Weinstein, and V. Zue, pp. 463-479 in Proc. Second DARPA Speech and Natural Language Workshop, Morgan Kaufmann, 1989.

10. "Multi-Site Data Collection for a Spoken Language System," MADCOW, *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

11. Polifroni, J., S. Seneff, V. W. Zue, and L. Hirschman, "ATIS Data Collection at MIT," *DARPA SLS Note 8*, Spoken Language Systems Group, MIT Laboratory for Computer Science, Cambridge, MA, November, 1990.

12. Polifroni, J., Hirschman, L., Seneff, S., and V. Zue, "Experiments in Evaluating Interactive Spoken Language Systems," *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

13. Price P., "Evaluation of Spoken Language Systems: The ATIS Domain," pp. 91-95 *in Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

14. Ramshaw, L.A. and S. Boisen, "An SLS Answer Comparator," *SLS Note 7*, BBN Systems and Technologies Corporation, Cambridge, MA, May 1990.

15. Seneff, S., Hirschman, L. and V. Zue, "Interactive Problem Solving and Dialogue in the ATIS Domain," pp. 354-359 *in Proc. Fourth Darpa Speech and Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

16. Shriberg, E., E. Wade, and P. Price, "Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction," *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

Proc. ICCS Workshop, Jan. 1993

The Relationship of Filled-Pause F0 to Prosodic Context

Elizabeth E. Shriberg

SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025 USA
and Department of Psychology, University of California at Berkeley

Robin J. Lickley

Centre for Speech Technology Research and Department of Linguistics
University of Edinburgh, 80, South Bridge, Edinburgh, EH11HN UK

1. ABSTRACT

Filled pauses in spontaneous speech present problems for models of speech understanding and automatic speech recognition. A potentially important cue to their recognition by both humans and machines is their typically low F0 [9, 7]. The current paper discusses results of a study [10] which sought to determine whether the F0 of filled pauses is relative to, or independent of, the F0 of surrounding lexical material. Clause-internal filled pauses and preceding peak F0 values for speakers of American and British English were examined. Higher peaks were found to be systematically associated with higher filled-pause values within speakers, supporting the "relative" hypothesis. In modeling this relationship it was found that a linear model, in which filled-pause F0 was expressed as an invariant (over speakers) proportion of the distance between the preceding peak F0 and a speaker-dependent terminal low F0, produced results nearly identical to those of a two-parameter model in which the coefficients of peak and terminal low F0 were allowed to vary freely. Analyses of additional variables showed the model to be less appropriate for filled pauses after sentence-initial peaks, but unaffected by temporal variables. These results suggest that clause-internal filled pauses, while lower in F0 than words in the message stream, nevertheless preserve information about the local prosodic context. Implications for psycholinguistics, speech recognition, and linguistic theory are discussed.

2. INTRODUCTION

Phenomena exhibited in spontaneous speech present new challenges for researchers in psychology, speech technology, and linguistics as the object of study shifts from carefully prepared "laboratory speech" to natural conversation. An important difference between spontaneous speech and speech that is read or rehearsed is that spontaneous speech is characterized by relatively high rates of hesitation pauses, repetitions and reformulations [3]. This paper examines one of the most common types of hesitation phenomena: the filled pause, usually realized orthographically as "um" or "uh."

Filled pauses can present problems for models of human language understanding and automatic speech recognition. In the case of human perception, what is remarkable is the extent to which filled pauses are "filtered out" in comprehension. Those familiar with the task of transcribing spontaneous speech will note that filled pauses are often missed in first passes at transcription; laboratory experiments [e.g., 5] have shown that listeners have difficulty locating filled pauses when monitoring for sentence content. In the case of speech recognition, filled pauses are problematic in that they are often misrecognized as words having similar phonetic features, such as "a", "an" or "and," or as syllables of longer words [1, 7, 9].

One source of information that is likely to be important in the successful perception and processing of spontaneous speech in general [see, for example, 6] and speech containing filled pauses in particular, is prosody. Recent work has contributed to our knowledge of the prosodic features of filled pauses. Studies of hesitations in a database of human-computer dialog [4, 11] show that filled pauses tend to occur in the lower region of a speaker's F0 range and have a level or falling tone [7], and, more specifically, that their F0 is typically lower than that of both accented and unaccented neighboring syllables [9].

For human perception, these findings may provide an account for the apparent perceptual separation of filled pauses from the message stream. The low F0 of filled pauses could aid automatic recognizers in distinguishing filled pauses from real words. In addition, linguists may be concerned with how to best represent these predictably low-F0 units in prosodic descriptions of spontaneous speech.

A question relevant to each of these areas concerns the nature of the relationship between the low F0 of filled pauses and the intonation of surrounding material. There are three possible relationships: 1) filled pauses may be produced at an absolute, speaker-specific F0 value regardless of their position within the sentence; 2) the F0 of filled pauses may vary within speaker, but the variation may be unpredictable; or 3) the F0 of filled pauses for a particular speaker may be predictable at better than chance, given knowledge about the prosodic context.

A study previously reported in [10] investigated the relationship between filled-pause F0 and intonational context; the current paper discusses results of that study in further detail. Since the question of interest concerned prosodic context, the relevant filled pauses to examine would be those that interrupt a prosodic phrase, as opposed to those that initiate a speaker's turn or occur between intonation phrases. The task of choosing filled pauses that occur within a prosodic phrase poses difficulties, however, in that: (1) it would be unclear how to label the data prosodically, since existing prosodic theories are not tailored to the description of material surrounding hesitation phenomena; (2) it is not clear what level of prosodic structure would be appropriate to use as the relevant unit for "interruption;" (3) choosing filled pauses on the basis of the prosody of surrounding material is potentially circular in that hesitations may themselves influence the prosody of that material; and (4) prosodic labeling requires listening to utterances and is time-consuming.

The scheme adopted was to study filled pauses that occurred within a syntactic clause. Filled pauses were considered to be "within-clause" if lexical material preceding the filled pause was syntactically incomplete, and strongly predicted continuation of the utterance after the filled pause. The value of the closest F0 peak preceding the filled pause was used as a measure of prosodic context, and the initial F0 value of the filled pause was used as a measure of filled-pause F0.

Within-clause filled pauses from speakers of American and speakers of British English, in two different discourse contexts, were examined to evaluate the three alternative hypotheses. The "absolute" hypothesis predicted that filled pauses would occur at a constant, speaker-dependent F0 value regardless of the value of the preceding peak F0. The "random" hypothesis predicted that filled-pause F0 values from a particular speaker would vary in a manner uncorrelated with preceding peak F0 values. The "relative" hypothesis predicted some form of systematic relationship between the peak and corresponding filled-pause F0 values.

3. METHOD

3.1. Subjects

Two quite different sets of data were analyzed. The first was a set of 120 clause-internal filled pauses from digitized utterances from 29 speakers (14 male, 15 female) of American English making air travel plans by speaking to a computer. The multi-site database is described in detail in [4]. The majority of examples came from "Wizard-of-Oz" systems, in which a human interpreted and responded to requests and thus "recognition" was perfect; a small number came from interaction with a Spoken Language System [11]. The number of clause-internal filled pauses per speaker used in the analyses ranged from 2 to 13; 82 of the examples came from 12 speakers (6 male, 6 female) having 5 or more examples each.

The second set consisted of 87 filled pauses taken from a corpus of six dialogues recorded digitally at the Department of Linguistics at the University of Edinburgh. Dialogues involved the second author and a colleague or acquaintance; they were natural, spontaneous conversations on various topics, with no set task. The subjects were 3 male and 3 female speakers of British English, without strong regional accents, who were unaware of the purpose of recording the conversations. The number of clause-internal filled pauses per speaker used in the analyses ranged from 6 to 28.

3.2. Filled Pauses

The goal of the study was to examine filled pauses that were likely to interrupt a prosodic phrase; however, because it would have been difficult and time-consuming to label the data sets prosodically in order to select the desired filled pauses, a method based largely on syntax was used. In general, the filled pauses selected for analysis were those that directly followed lexical material that would have been syntactically incomplete if the utterance had not continued after the filled pause. It was felt that this would be an efficient, straightforward, and easy-to-replicate method for capturing many of the filled pauses that did interrupt prosodic phrases, while avoiding the complex and time-consuming task of prosodic labeling. Some examples from the American data set are listed in Table 1.

Table 1: Examples of Clause-Internal Filled Pauses

Incomplete	"Looking for"	Example
NP	N	...the lowest [uh] fare...
VP (trans)	NP	...book [uh] the flight...
PP	NP	...leave at [um] noon...
AUX	S	Does [uh] Delta fly...

The researchers tried to determine whether or not a listener would feel it was possible that the speaker could have ended an utterance before the filled pause, based on a transcription alone, but taking semantic and pragmatic information into account. For example, filled pauses in utterances such as:

Show me flights flying [uh] from Boston.

in which material before the filled pause is not necessarily syntactically incomplete, but which would seem incomplete to a listener given the discourse context, were included in the analyses.

Conversely, some utterances which could be viewed as meeting the syntactic expectancy requirement were not included in the analyses. These were cases in which the only item preceding the filled pause in the same clause was a conjunction such as "and" or "but.," a lexical filler such as "well" or "okay," or another filled pause. Such cases were excluded because of the higher likelihood of a prosodic boundary immediately preceding the filled pause.

3.3. Apparatus

The digitized waveforms were sampled at 8 or 16 kHz and all waveforms and pitch tracks were examined using the Entropic ESPS/Waves+ software on a Sun 4 workstation.

3.4. Procedure

The American and British data were coded independently by the first and second authors, respectively. For each within-clause filled pause having reliable pitch tracks, the researcher recorded five F0 values, four measures of duration, and values for four additional variables.

The F0 of each filled pause was measured at both the beginning and end of the filled pause. These values describe the F0 of filled pauses well, since most fall fairly linearly. Analyses in the present work used the initial filled-pause F0 as a measure of filled-pause F0. F0 was also recorded at the F0 peaks most closely preceding and following the filled pause; results reported here used only the preceding peak as a measure of prosodic context. Alternative measures of context (for example topline, or preceding low accents) could also be used, but could be more difficult to measure and locate than F0 peaks. Peak values were restricted to occur on words within the clause containing the filled pause. In most cases, the peak was marked on a syllable perceived to be accented; in a few cases no accented syllable was available and the highest preceding F0 value was used.

A fifth F0 value, which will be referred to as the "terminal low F0," was measured after final lowering in a manner similar to that described in [2]; i.e. for utterances containing a terminal fall, F0 was measured at the lowest point in the fall, disregarding regions associated with errors in pitch tracking or vocal fry. The purpose of this measure was to provide a single, stable, speaker-dependent F0 value for each speaker. The underlying assumption in the present work was that this value should correspond to a speaker's lowest possible F0, as opposed to the lowest F0 realized in any particular utterance, since the former would be the more stable value given the inherently positively skewed distribution of terminal low F0 values. Therefore, terminal low F0 values were obtained for all utterances for a particular speaker that contained a terminal fall. The lowest of these values was then used as the estimate of the speaker's terminal low F0 for all speech tokens from that speaker in the analyses. Care was taken to assure that the lowest terminal F0 value did not appear to be an outlier when compared with the other terminal F0 values obtained for the same speaker.

Four measures of duration were recorded, including the duration of the filled pause, that of preceding and following silent hesitation pauses (if any), and that of the time (and also the number of syllables) between the preceding peak and the beginning of the filled pause.

Values for additional variables of interest were also recorded, including the sex of the speaker, whether or not the filled pause preceded a repetition, repair, or fresh start, whether or not the preceding peak was marked on a sentence-initial accent, and whether the filled pause was "um" or "uh."

4. RESULTS

Figures 1-4 show data for a male or female speaker from each of the data sets (American and British). Time-normalized F0 values are shown for the preceding peak F0, initial filled-pause F0, final filled-pause F0, and following peak F0 in multiple examples of filled pauses for the particular speaker. Each speaker's estimated terminal low F0 is also indicated.

4.1. Testing the Hypotheses: Sign Test

The first thing to note about the plots is that, in general, the drop to the filled pause from the preceding peak scales with the peak values, so that higher peaks tend to have higher following filled pauses. This simple assumption was tested using data from all 35 speakers. The highest and lowest preceding peak F0 values over all examples from a particular speaker were extracted and the associated filled pause values compared in a Sign test. In 34/35 cases, the higher preceding peak value was associated with a higher filled pause value, $p < .0001$. This highly significant result is consistent with the relative hypothesis and inconsistent with the absolute and random hypotheses.

4.2. Modeling the Relationship

A second observation about Figs. 1-4 is that there appears to be a lower bound of F0: filled pauses do not seem to go below the terminal F0. This suggests that filled-pause F0 cannot be expressed as a simple subtractive function of

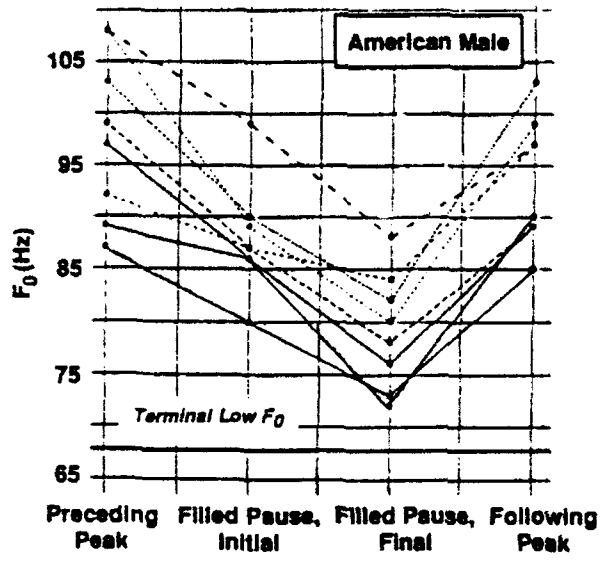


Figure 1: Peak and Filled-Pause F0 for American Male

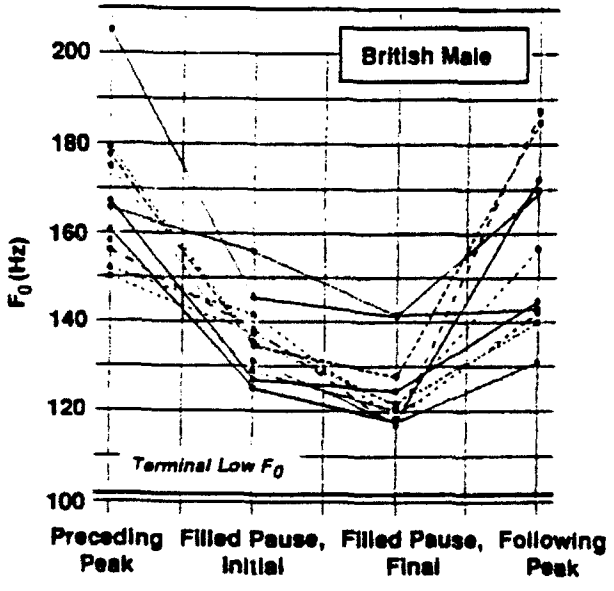


Figure 2: Peak and Filled-Pause F0 for British Male

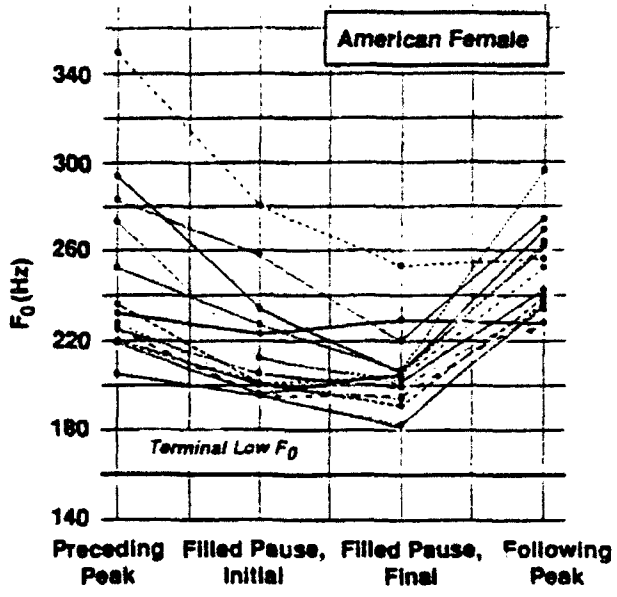


Figure 3: Peak and Filled-Pause F0 for American Female

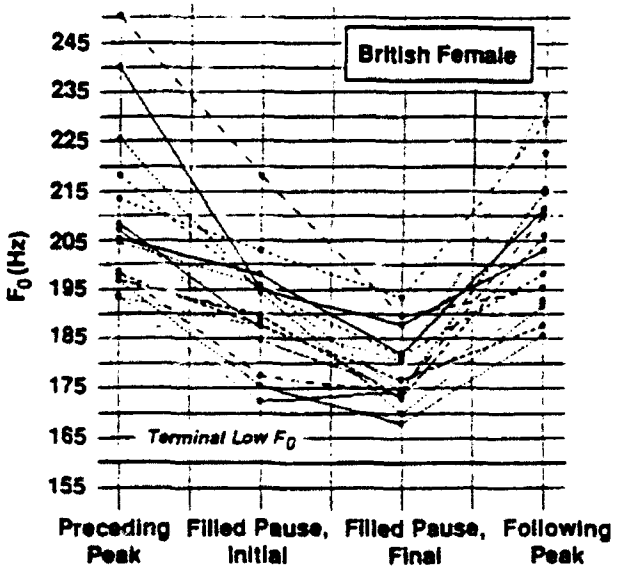


Figure 4: Peak and Filled-Pause F0 for British Female

peak F0. A third observation is that there seems to be a compressive effect for peaks closer to the terminal F0, with lower peaks producing less of a drop to the filled pause than higher ones. This observation suggests that filled-pause F0 cannot be expressed as a simple multiplicative function of peak F0, since such a function would predict parallel curves. Exceptions to this trend are the filled pauses following the very highest peak examples in Figs. 1, 2, and 4, which do not drop as far as expected. However, these examples form a special class; they correspond to filled pauses following peaks marked on sentence-initial accented syllables which, as discussed later, appear to behave differently from other clause-internal filled pauses.

Based on these observations, we proposed a simple linear model, in which filled-pause F0 (F0 fp) is the F0 value occurring at a fixed proportion of the distance between the peak F0 (F0 peak) and the terminal low F0 (F0 min):

$$F_{0\,fp} = r\,(F_{0\,peak} - F_{0\,min}) + F_{0\,min}$$

This is a single-parameter model, since the coefficients of peak F0 and terminal low F0 are both determined by r.

We determined the value of r empirically for each filled pause token from the set of American and British speakers with five or more examples each (18 subjects, 169 filled

pauses.) Means for tokens broken down by American/British and male/female are shown in Table 2.

Table 2: Values of r

Subject	# of speakers	# of tokens	Mean r	s.d. of r
American				
male	6	39	.596	.214
female	6	43	.626	.158
British				
male	3	55	.607	.240
female	3	32	.636	.242

Because results for the American and British data were remarkably similar, data were pooled for all further analyses. Although the value of r appears to be slightly higher for women in both groups, the differences are nonsignificant (as can be seen by comparing them to the magnitude of the standard deviations.)

A linear regression with the constant term suppressed, performed using the raw data from subjects represented in Table 2, and using the mean r determined over the entire set (0.62), yielded a standard error in prediction of 15.41 Hz. A comparison of this model to two other linear models is shown in Table 3. Investigation of higher-order models was not warranted given the lack of evidence for a nonlinear relationship, and the potential danger of over-fitting the small data set at hand. The proposed model was clearly better than one in which only the peak was used to predict the filled pause F0. It was also remarkably close in prediction accuracy to results produced by a two-parameter model which allowed the coefficients of peak and terminal low F0 to vary freely

Table 3: Comparison of Models

Variables	# of Parameters	RMS error (Hz)
peak, terminal low F0	1	15.41
peak	1	19.58
peak, terminal low F0	2	15.25

4.3. Optimal Reference F0

An issue addressed was whether, given the proposed model, the estimated terminal low F0 values used corresponded to the optimal reference F0 values for prediction. Ideally, regressions solving for the optimal r and constant for each speaker would allow for comparison of these results to those obtained using the observed terminal low values, however, to be meaningful such analyses require more data per speaker. Nevertheless, analyses performed for a subset (N=6) of the 18 subjects who had the largest numbers of examples revealed that in each case the optimal reference F0 was higher than the observed terminal low F0. Therefore a number of modifications of the observed values in the 18-speaker data set were computed. For each modification, r was redetermined using the new terminal low values, and filled pauses were predicted using the new, overall average r and new low F0 values. It was found that the minimum standard error (15.16 Hz, as opposed to 15.41 Hz for the original terminal low values) was produced when observed terminal low values were increased by roughly 10%.

4.4. Effect of Duration

There was no correlation between the time or the number of syllables from the peak to the filled pause and the drop size. As shown in Figure 5, the drop in F0 from the preceding peak to the filled pause did not seem to depend on the amount of time elapsed between these two points

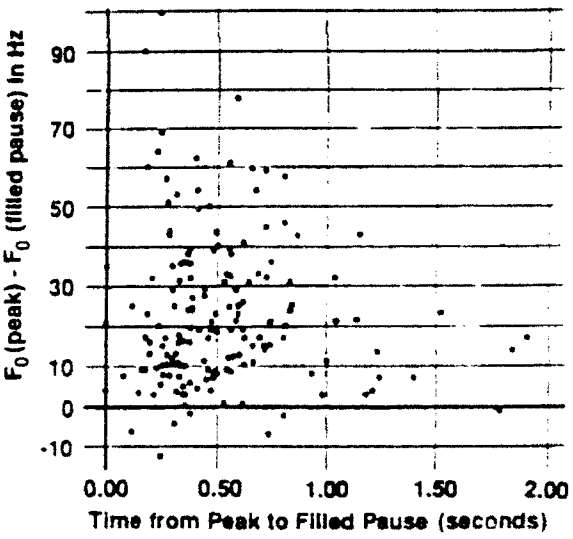


Figure 5: Effect of Time from Peak on F0 Drop

In addition, there did not seem to be any relationship between the duration of the filled pause itself and the size of the fall in F0 over the course of the filled pause, as shown in Figure 6

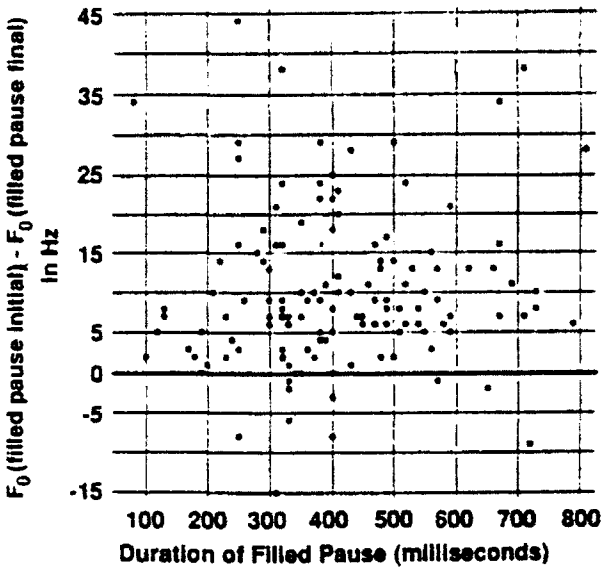


Figure 6: Effect of Filled-Pause Duration on Filled-Pause Fall

4.5. Effect of Additional Variables

Results of regressions performed using the observed terminal low F0 values and selecting independently for values of additional variables are shown in Table 4.

Table 4: Effect of Additional Variables

Data in Analysis	RMS error (Hz)	# of tokens
all data	15.41	169
male speaker	12.36	94
female speaker	18.42	75
peak on sentence-initial accent	30.30	26
peak not on sentence-initial accent	10.90	143
no other disfluency present	14.36	141
filled pause precedes repetition	23.90	11
filled pause precedes replacement	13.09	7
filled pause precedes fresh start	17.90	9
filled pause is "um"	15.29	86
filled pause is "uh"	15.20	83

As can be seen, the factor most influencing prediction accuracy was whether or not the preceding peak was marked on a sentence-initial accented syllable. Although conclusions cannot be drawn given the small number of tokens of this type, it is worth noting that the error in prediction was always in the same direction, with the actual filled pause occurring at a higher F0 value than predicted by the model. Tokens not involving disfluencies had a lower standard error than that observed overall;, however, results for the different types of disfluencies were inconclusive due to small sample size. Prediction error was not affected by whether the filled pause was "um" or "uh" (although "um" tokens were significantly longer in duration than "uh" tokens, and it should be borne in mind that the present model predicted only the initial F0 of the filled pause.) Prediction accuracy was also not affected by the sex of the speaker; that females had a higher standard error than males was expected given the roughly 50% higher terminal low F0 values for the females.

5. DISCUSSION

5.1. Evaluation of Hypotheses

Two different sets of spontaneous speech data were examined to explore the relationship between the F0 of clause-internal filled pauses and their surrounding context. Results show that the initial F0 of clause-internal filled pauses scales with the F0 of preceding peaks, strongly supporting the "relative" hypothesis.

5.2. Modeling the relationship

Inspection of data from individual subjects revealed that in addition to the scaling of filled pause F0 with preceding peak F0, there was also a lower bound of filled-pause F0 values, and a compressive effect on the size of the drop from the preceding peak to the filled pause as peaks approached the lower portion of a speaker's range.

A model of filled-pause F0 was proposed to reflect these observations. The model was not necessarily intended to have any theoretical interpretation, but rather simply to predict the value of filled-pause F0 using other accessible values of F0. Filled-pause F0 was expressed as a function of three values: (1) a speaker-dependent fixed terminal low F0 value (representing the speaker); (2) the value of the preceding peak F0 (representing the particular prosodic context); and (3) a fixed, speaker-independent scaling factor, r (to express the relationship between the two previous values and filled-pause F0). This is an extremely constrained model, with only one free parameter (r). In addition, the constant term in the model corresponds to a speaker's empirically measured terminal low F0, as opposed to some

F0 value unrelated to prosodic phenomena (for example one outside the speaker's range). Clearly, the current model could also be rewritten to be expressed using coordinates related to a different model (for example, a declination model); the present model is at least as parsimonious as any alternative model in which the functions rewriting peak and terminal low F0 in terms of other variables are linear.

One certainly cannot draw conclusions about the appropriateness of models based on examination of the limited set of data used in the present study. Nevertheless, it is impressive how well the proposed model was able to predict the data. Of possible linear models (there was no evidence for a nonlinear relationship when data from individual subjects were examined) the present model performed extremely well, producing results only very slightly less accurate than a linear model with an additional parameter (in which the coefficients of peak and terminal low F0 were allowed to vary freely.) Real evidence in support of a model such as the present one, however, will probably have to come from comparison of r in the present model to scaling factors proposed in studies of other prosodic phenomena, for example low-tone scaling or the scaling of parentheticals.

5.3. Values of r

It was found that the average value of the parameter r, which expresses the proportion of the distance from terminal low F0 to peak F0 at which filled-pause F0 occurs, did not differ across the American and British data sets. This suggests that the intonation of clause-internal filled pauses, at least as measured by the relationship between preceding peak F0 and initial filled-pause F0, may be independent of factors such as dialect and discourse setting. Mean r values also did not differ across sex. Since speaker sex is highly correlated with the terminal low F0, this lack of a difference in r between sexes is consistent with the appropriateness of a linear model.

5.4. Optimal Reference F0

The value of terminal low F0, a speaker-dependent variable corresponding to the lowest observed F0 value produced after a terminal fall, was found to be slightly lower than the value which optimized prediction. The overall standard error over the data set was slightly decreased when the value of terminal low F0 was raised by 10% for each speaker. A larger data set, with more tokens per speaker, is needed in order to further investigate this finding; it suggests, however, that the value used to scale pitch over the course of an utterance is higher than the F0 measured after final lowering. This is consistent with proposals in the literature [e.g., 8], although it does not distinguish between a declination model and one in which F0 falls abruptly at the end of an utterance. It should be noted that the decision to use the lowest observed terminal low F0, as opposed to other possible values (for example, the mean of all observations) was made because the aim was to get a stable estimate for each speaker, given a positively skewed distribution of low F0 values. Using values such as the mean would therefore be inappropriate. That is, by using mean low F0, one cannot improve results in a principled way, whereas by using a stable estimate such as minimum low F0 (assuming however that there are enough observations available to adequately estimate this value), one can examine the relationship between minimum low F0 and the F0 that optimizes prediction. For exploratory purposes, however, an analysis using mean low F0 values was performed post hoc on the present data set. Results showed a marked reduction in prediction accuracy, and a distribution of r values with much higher standard deviations. Nevertheless, it is conceivable that an analysis using mean low F0 values on a different set of data could produce better results than an analysis using minimum F0 values; such a result would not be meaningful, however, but would rather be due to the fact that mean low F0, like optimal reference F0, is higher than minimum low F0.

5.5. Effect of Duration

Results also suggest that the intonation of filled pauses may be independent of temporal variables. As shown in Fig. 5, there was no correlation between the size of the drop in F0 from the preceding peak to the filled pause and the distance (in time or syllables) between these points; i.e. filled-pause F0 was unrelated to whether or not words and/or silent pauses intervened between the preceding peak and the filled pause. Also, rather surprisingly, there was no correlation between the duration of the filled pause and how far in F0 it fell, as shown in Fig. 6. Most clause-internal filled pauses have a slight linear fall; the fact that longer filled pauses do not fall to a lower F0 than shorter filled pauses implies that the longer tokens either start out with a shallower falling slope, or that they level off in F0 once they reach a point that is "too low" for the local prosodic range. It is also possible that for long hesitations, speakers may stop the filled pause completely and use a silent pause when they have dropped too far. Future work will attempt to examine these issues more closely. These results add further support to the notion that clause-internal filled pauses are in some sense "well-formed" since the range of F0 values for a filled pause is determined by the local prosodic context. In addition, these findings suggest that prosodic regularities in filled pauses may be found more in F0 than in duration measures; this possibility seems reasonable because hesitations, by definition, interrupt the temporal course of production.

5.6. Effect of Sentence-Initial Peaks

As shown in Table 4, prediction error of the proposed model was much greater for filled pauses following peaks marked on sentence-initial accents than for filled pauses elsewhere. In each case following a sentence-initial peak, the prediction of the model for filled-pause F0 was lower than the

observed value; when this relatively small set of tokens was removed from the analyses, the overall error in prediction was reduced substantially. This finding is consistent with the notion that the F0 of filled pauses preserves information about the current prosodic context: filled pauses after peaks corresponding to extra-high sentence-initial accents are themselves extra-high.

5.7. Implications for Areas of Research

The finding that the F0 of filled pauses is relative to prosodic context has implications for models of human speech perception, automatic speech recognition, and for theoretical and descriptive studies of prosody.

The low F0 of filled pauses may help explain why listeners have trouble locating them with respect to words in the message stream; low F0 may also contribute to listeners' ability to filter out filled pauses in comprehension. Experiments designed to test these hypotheses, by using resynthesis to "lift" filled pauses up to the F0 of the region of the lexical material in an utterance, will be conducted in future work. These tests predict that raising the F0 of filled pauses will facilitate listeners' ability to locate them, and also possibly impair comprehension. The finding that the F0 of filled pauses is relative to prosodic context suggests that speakers may attempt to preserve the current prosodic range when hesitating, possibly to inform the listener that they intend to continue where they left off, rather than to abandon a portion of the utterance preceding the filled pause. Thus, a question to be pursued in further work is whether there is a difference between filled pauses that interrupt otherwise fluent clauses, and those that occur at the interruption point of a repair or before a fresh start, since in the latter cases the speaker is abandoning previous material. There were not enough examples of filled pauses in repairs or fresh starts in the present data set to address this question; however preliminary results of additional data suggest that very brief filled pauses, which fall rapidly in F0, often mark a repair (but these are not necessary features for the marking of a repair), and that an unexpectedly high F0 on a filled pause seems to be a very good indicator of a fresh start (essentially an F0 "reset" to begin a new utterance after the filled pause).

Speech recognition systems may be able to take advantage of predictably low F0 in spotting filled pauses. In order to do so successfully however, at least in the case of filled pauses within a clause, these systems will need to take into account the intonation of the local context, rather than using absolute speaker-specific F0 values. Spoken language systems may also benefit from knowing more about prosodic differences between filled pauses in different syntactic environments. Preliminary analyses suggest that whereas clause-internal filled pauses nearly always have a low and falling F0, filled pauses that occur turn-initially or between sentences often have a higher and level or even slightly rising F0. Such information should aid attempts to recognize filled pauses; in addition the recognition of filled pauses having these different prosodic characteristics could contribute information about sentence structure for natural language processing.

As linguists move from the study of read or rehearsed speech to spontaneous discourse, it should become increasingly important for them to consider the prosody of disfluencies, since as shown in the present study, some phenomena considered to be disfluent may exhibit prosodic regularities. This work also suggests that in the case of clause-internal filled pauses, F0, rather than duration, may be the most important prosodic feature to explore. It should prove useful for linguists to include methods for annotating disfluencies in systems developed for the prosodic labeling of spontaneous speech.

6. CONCLUSION

This work has shown that the F0 of one type of speech disfluency, the clause-internal filled pause, is related to the intonation of surrounding material in the message stream. Further work in this area could enhance our knowledge of the production and processing of spontaneous speech, help us learn how to apply these findings to aid speech recognition, and encourage the consideration of hesitations and other disfluencies in theoretical and descriptive work on prosody.

ACKNOWLEDGMENTS

We wish to thank Mark Anderson for helpful discussions on the modeling of F0, and John Bear and Beth Ann Hockey for suggestions regarding syntactic-based principles for categorizing filled pauses. The research of the first author was supported by the Defense Advanced Research Projects Agency under Contract ONR N00014-90-C-0085 with the Office of Naval Research, and also by NSF Grant IRI-890529 from the National Science Foundation. The second author was supported by Award number 87310722 from the UK Science and Engineering Research Council. The opinions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

1. Butzberger, J., H. Murveit, E. Shriberg, & P. Price, "Spontaneous Speech Effects in Large Vocabulary Speech Recognition Applications," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

2. Liberman, M. & J. Pierrehumbert, "Intonational Invariance under Changes in Pitch Range and Length," *Language Sound Structure*, M. Aronoff and R. Oehrle (eds.), MIT Press, 1984.

3. Maclay, H. & C. Osgood, "Hesitation Phenomena in Spontaneous English Speech," *Word*, 15, pp. 19-44, 1959.

4. MADCOW, "Multi-Site Data Collection for a Spoken Language Corpus," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

5. Martin, J., and W. Strange, "The Perception of Hesitation in Spontaneous Speech," *Perception & Psychophysics*, 3, pp. 427-38, 1968.

6. Nooteboom, S., P. Brokx & J. De Rooij, "Contributions of Prosody to Speech Perception," *Studies in the Perception of Language*, W. Levelt and F. D'Arcais (eds.), John Wiley and Sons, 1978.

7. O'Shaughnessy, D., "Recognition of Hesitations in Spontaneous Speech." *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 521-524, 1992.

8. Pierrehumbert, J., "The Phonology and Phonetics of English Intonation," Ph.D. dissertation, MIT, Cambridge, MA, 1980.

9. Shriberg, E., "Intonation of Filled Pauses in Spontaneous Speech." Paper presented at the Conference on Grammatical Foundations of Prosody and Discourse, July 5-6, Santa Cruz, 1991.

10. Shriberg, E. & Lickley, R. "Intonation of Clause-Internal Filled Pauses. *Proceedings of the International Conference on Spoken Language Processing*, 1992.

11. Shriberg, E., E. Wade & P. Price, "Human-Machine Problem Solving Using Spoken Language Systems (SLS). Factors Affecting Performance and User Satisfaction," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992

Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction

Elizabeth Shriberg[1,3], *Elizabeth Wade*[2,3], *Patti Price*[3]

[1]University of California at Berkeley, Department of Psychology, Berkeley, CA 94720
[2]Stanford University, Department of Psychology, Stanford, CA 94305
[3]SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94303

ABSTRACT

We have analyzed three factors affecting user satisfaction and system performance using an SLS implemented in the ATIS domain. We have found that: (1) trade-offs between speed and accuracy have different implications for user satisfaction; (2) recognition performance improves over time, at least in part because of a reduction in sentence perplexity; and (3) hyperarticulation increases recognition errors, and while instructions can reduce this behavior, they do not result in improved recognition performance. We conclude that while users may adapt to some aspects of an SLS, certain types of user behavior may require technological solutions.

1. INTRODUCTION

Data collection is a critical component of the DARPA Spoken Language Systems (SLS) program. Data are crucial not only for system training, development and evaluation, but also for analyses that can provide insight to guide future research and development. By observing users interacting with an SLS under different conditions, we can assess which issues may best be addressed by human factors and which will require technological solutions. System developers can benefit from considering not only initial use of an SLS, but also the experience of a user over time.

Systems based on current technology work best when speech and language closely resemble the training data used to develop the system. However, there is considerable variability in the degree to which the speech and language of new users match that of the training data. The current paper examines the importance of this initial match. It is possible that users whose speech does not conform to the system may be able to adapt their behavior over time (e.g., Stern and Rudnicky [11]). In order to evaluate technology in terms of the demands of the application, we need to understand the extent and the nature of such adaptation and the conditions that affect it. Although system performance can be measured in a number of ways, in this paper, we focus on (1) self-reports of user satisfaction, and (2) recognition performance. Further studies could include additional measures.

SRI has been collecting data in the air travel planning domain using a number of different systems (see Bly et al. [1]; Kowtko and Price [5]). In moving from wizard-based data collection to the use of SRI's SLS, we observed changes in user behavior that were associated with system errors. Some of these behaviors were adaptive; for example, learning to avoid out-of-vocabulary words or unusual syntax should facilitate successful interaction. Other behaviors, however, were non-adaptive and could actually impede the interaction. For example, speaking more loudly or in a hyperarticulate style may be detrimental to system performance insofar as these styles differ from those observed in training material dominated by wizard-mediated data in which system errors are minimal.

It is difficult to predict how well an SLS will need to perform in order to be acceptable to users. Both speed and accuracy are crucial to system acceptability; we have therefore collected data using versions of the system that prioritize one of these parameters at the expense of the other. The present study first addresses the issue of user satisfaction with different levels of system speed and accuracy and then focuses on an example of an adaptive behavior and another that is maladaptive. These behaviors represent a subset of potential factors influencing human-machine interaction. Because these issues are not restricted to any particular system, they should be of general interest to developers of SLS technology.

In the first study, we compared three points in the speed-accuracy space for this application: (1) an extremely slow but very accurate wizard-mediated system (described in Bly et al. [1]) with a 2-3 minute response time and a minimal error rate; (2) a software version of the DECIPHER recognizer with a response time of several times real time and a fairly low word error rate; and (3) a version of the DECIPHER recognizer implemented in special-purpose hardware using older word models, which has a very fast response time but currently has a higher word error rate.

We compared user satisfaction based on responses to a post-session questionnaire.

The second study investigated the effect of user experience on syntax and word choice. We hypothesized that one way users might adapt would be to conform to the language models constraining recognition. We therefore measured recognition performance in subjects' first and second scenarios, and compared sentence perplexities in order to determine whether any changes in recognition performance could be attributed to a change in perplexity.

The third study examined the effect of hyperarticulate speech on recognition and tested whether instructions to users could reduce this potentially maladaptive behavior. We coded each utterance for hyperarticulation and compared recognizer performance for normal and hyperarticulate utterances. We also compared rates of hyperarticulation for subjects who were either given or not given the instructions.

2. DATA COLLECTION METHODS

2.1. Subjects

Data from a total of 145 subjects were included in the analyses. Subsets of these data were chosen for inclusion in each analysis in order to counterbalance for gender and scenario. The majority of subjects were SRI employees recruited from an advertisement in an internal newsletter; a small number were students from a nearby university, employees in a local research corporation, or members of a volunteer organization. Subjects were native speakers of English, ranged in age from 22 to 71 and had varying degrees of experience with travel planning and computers.

2.2. Materials

Four different travel-planning scenarios were used. One entailed arranging flights to two cities in three days; a second entailed finding two fares for the price of a first class fare; a third required coordinating the arrival times of three flights from different cities; and a fourth involved weighing factors such as fares and meals in order to choose between two flight times. Because the task demands of the scenarios were different, we controlled for scenario in the analyses.

2.3. Apparatus

The data were collected using two versions of SRI's SLS (with no human in the loop); the first study also included data collected in a Wizard of Oz setting (Bly et al. [1]). The basic characteristics of the DECIPHER speech recognition component are described in Murveit et al.[7,9], and the basic characteristics of the natural language understanding component are described in Jackson et al. [4]. Some subjects used the real-time hardware version of the DECIPHER system (Murveit and Weintraub [8]; Weintraub et al. [12]); others used the software version of the system, which was a modified version of SRI's benchmark system (as described in the references above) tuned using the pruning threshold to improve speed at the cost of introducing a small number of recognition errors.

SRI's SLS technology was implemented in the air travel planning domain, a domain with which many people are familiar (see Price [10]). The underlying database was a relational version of an 11-city subset of the Official Airline Guide. Two DARPA/NIST standard microphones were used: the Sennheiser HMD-410 close-talking microphone and the Crown PCC-160 table-top microphone. Most data were collected with two channels; some of the early data were collected using only the Sennheiser microphone. When both microphones were used, recognition was based on the Sennheiser input.

The interface presented the user with a screen showing a large button labeled "Click Here to Talk." A mouse click on this button caused the system to capture speech starting a half second before the click; the system automatically determined when the speaker finished speaking based on silence duration set at a threshold of two seconds. The user could move to the context of previous questions via mouse clicks. Once the speech was processed, the screen displayed the recognized string of words, a "paraphrase" of the system's understanding of the request, and, where appropriate, a formatted table of data containing the answer to the query. In cases where the natural language component could not arrive at a reasonable answer, a message window appeared displaying one of a small number of error messages. A log file was automatically created, containing time-stamps marking each action by the user and by the system.

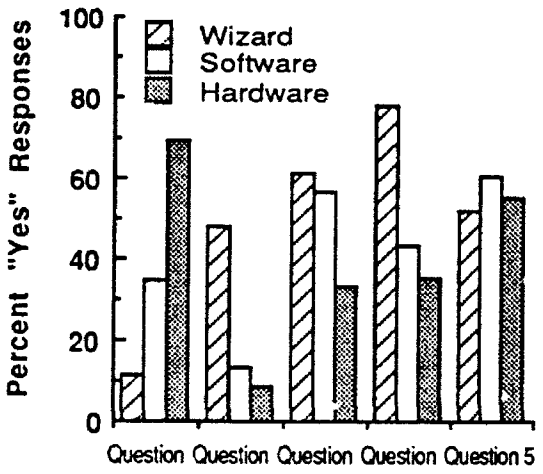
2.4. Procedure

Subjects were seated in a quiet room in front of a color monitor, and had use of a mouse and microphone(s) but no keyboard. They were given a short demonstration on how to use the system. Some of the subjects were given additional instructions explaining that, while they might have a tendency to enunciate more clearly in the face of recognition errors, they should try to speak naturally, since the system was not trained on overenunciated or separated speech. Once subjects were comfortable with the system, they were left alone in the room while they solved travel planning scenarios. After they finished as many scenarios as possible within an hour, they were asked to fill out a questionnaire and were given a choice of gift certificate for use at a local bookstore or a contribution to a charitable institution.

3. EXPERIMENTS

3.1. The Effects of Speed and Accuracy Trade-offs on User Satisfaction

Since in general, speech understanding systems can trade accuracy for speed, we first assessed how these parameters might affect user behavior and acceptance of the system. The software version of the recognizer was slower than the hardware version (2.5 compared to 0.42 times the utterance duration), but was substantially more accurate (with a word error rate of 16.1% a- compared with 24.8% on the same sound files).



1. Were the answers provided quickly enough?
2. Did the system understand your requests the first time?
3. I focused most of my attention on solving the problems, rather than trying to make the system understand me.
4. Do you think a person unfamiliar with computers could use the system easily?
5. Would you prefer this method to looking up the information in a book?

Figure 1: User Satisfaction

To assess user satisfaction, we compared questionnaire responses for 46 subjects who used the hardware, 23 who used the software, and 46 who used the earlier wizard-mediated system. Mean responses are shown in Figure 1. In general, user satisfaction with the speed of the system correlated with the response time of the system they used; when asked, "Were the answers provided quickly enough?" 69.6% of the hardware users responded "Yes." In contrast, only 34.8% of the software users and a mere 11.1% of the wizard-system users gave "Yes" responses, a significant difference from the hardware result, χ^2 (df=4) = 35.6, $p < .001$. Although hardware users were pleased with the speed of the system; they were less likely than wizard system and software users to say they focused their attention on solving the problem rather than on trying to make the system understand them (33.3% as compared with 61.4% and 56.5%, respectively), a marginally significant effect, χ^2 (df=4) = 7.8, $p < .10$.

On several other measures users found the wizard-based system preferable to either the software or the hardware. More wizard-system users said that the system usually understood them the first time (47.8% as compared with 13.0% and 8.7% for the software and hardware users, respectively), χ^2(df=4) = 22.5, $p < .001$. Overall, the wizard system users were more likely to say the system could be easily used by a person who was unfamiliar with computers (78% compared with 43.5% and 35.6% for the software and hardware, respectively) χ^2 (df=4) = 20.5, $p < .001$. However, in terms of general satisfaction, as expressed in whether the subjects said they would prefer using the system to looking the information up in a book, there was no significant difference between the groups, with 52.3%, 60.9% and 55.6% "Yes" answers for the three groups respectively.

Because the hardware system was least satisfying to users in terms of recognition accuracy, we concluded that the hardware would provide the greatest potential for user adaptation to the system. For this reason, we used the hardware system to collect data on the effects of user experience and instructions regarding hyperarticulation.

3.2. Effect of User Experience on Recognition

User experience was evaluated in a within-subjects design, counterbalanced for scenario, that compared 24 users' first and second sessions. As a global measure of adaptation, we looked at how long it took subjects to complete their two scenarios. Although subjects were not told to solve the scenarios as quickly as possible, they nevertheless took less time (10.5 compared to 13.0 minutes) to complete their second scenarios, $F(1,23) = 5.78$, $p < .05$. This difference was partially but not completely attributable to a lower number of total utterances in the second scenario.

The users also elicited fewer recognition errors in the second scenario. The mean word error rate was 20.4% for the first scenario but fell to 16.1% for the second, $F(1,22) = 5.60$, $p < .05$. However, not all users decreased their recognition error rate. There was a significant interaction between initial error rate and change in error rate from the first scenario to the second, $F(1,22) = 10.98$, $p < .01$. Subjects who had recognition error rates of 20% or worse in the first scenario (N=11) tended to improve recognition performance, while subjects who had better initial performance (N=13) did not (Figure 2). Subjects with initial error rates of 20% or

higher went from an average of 31.3% errors down to 19.6%, while subjects with initially lower error rates showed no statistically significant change. For those subjects who did improve recognition performance, the improvement could only be due to user adaptation, since the same SLS version was used for both scenarios.

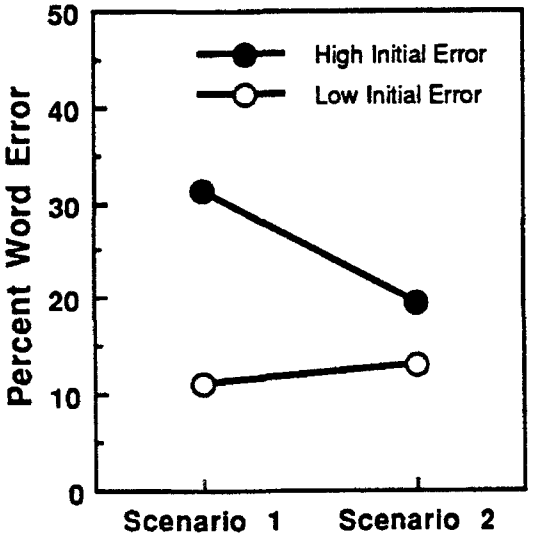


Figure 2: Recognition accuracy over time.

The improvement in recognition may be due in part to user adaptation to the language models used. As a measure of deviation from the system's language models, we used test-set perplexity, which was based on the bigram probabilities of the observed word sequences. As would be expected, there was a significant, positive average correlation between utterance word error and perplexity: mean $r = .28$, $t = 4.55$, $p < .001$. Thus, one way for subjects to improve recognition accuracy would be to change their language to conform to that of the system model. Perplexity may therefore play a role in the decrease in recognition error rates observed over time for those subjects who had an error rate of 20% or worse in their first scenario. For this group of subjects, there was a tendency to produce queries with lower sentence perplexity in the second scenario (Figure 3). Using the median as a measure of central tendency (a more stable measure due to the inherent positive skew of perplexity), we found that the average median sentence perplexity was 25.3 for the first scenario and 19.4 for the second, a reliable difference, $F(1,10) = 7.44$, $p < .05$.

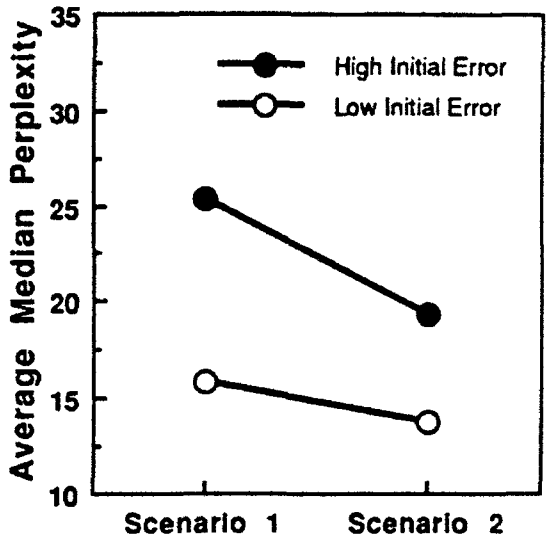


Figure 3: Median perplexity over time.

In addition to decreasing perplexity, subjects who had initial error rates of greater than 20% also tended to decrease the use of out-of-vocabulary words in the second scenario, whereas subjects who had lower error rates did not, a significant interaction, $F(1,22) = 6.10$, $p < .05$. Overall, however, the use of out-of-vocabulary words was rare.

These findings indicate that at least to some degree, subjects adapted to the language models of the system and, in doing so, managed to improve the recognizer's performance. Quite possibly, subjects were finding ways to phrase their queries that produced successful answers, and then reproducing these phrases in subsequent queries. In future work, further analyses (for example, looking at dialogue) will address this issue in greater detail.

3.3. Effect of Instructions on Speech Style

Another potential source of recognition errors arises when the speech of the user deviates from the acoustic models of the system. Since the vast majority of the data used to train the DECIPHER recognizer came from wizard-mediated data collection [6], where recognition performance was nearly perfect, examples of "frustrated" speech were rare. In human-human interaction, when an addressee (such as a foreigner) has difficulty understanding, speakers change their speech style to enunciate more clearly than usual (Ferguson [3]). We suspected that a similar effect might occur for people speaking to a machine that displayed feedback showing less than perfect understanding. We noticed that, when using an SLS as opposed to a wizard-mediated system, subjects tended to hyperarticulate: releasing stops, emphasizing initial word segments, pausing between words, and increasing vocal effort.

Although hyperarticulation is a multifaceted behavior, it was nevertheless possible to make global judgments about individual utterances. Hyperarticulation was coded for each utterance on a three-point scale by listening to the utterances. Utterances were coded as (1) clearly natural sounding, (2) strongly hyperarticulated, or (3) somewhat hyperarticulated. The coding was done blindly without reference to session context or system performance.

Using a within-subjects design, so that any differences in recognition performance could be attributed to a change in speech style, rather than speaker effects, we analyzed the speech style of 24 subjects' first scenarios (future analyses will also examine repeat scenarios). These subjects (of whom 20 were also included in the previous analysis of user experience) all used the hardware system. The subjects averaged about 10 natural sounding, 4 somewhat hyperarticulate, and 5 strongly hyperarticulate utterances each. For the 13 subjects who had at least three natural and three strongly hyperarticulated utterances, we compared recognition performance within subjects and found that the strongly hyperarticulate utterances resulted in higher word error rates, $F(1,12) = 5.19$, $p < .05$.

Hyperarticulation was reduced, however, by giving users instructions not to "overenunciate" and by explaining that the system was trained on "normal" speech. We calculated a hyperarticulation score for each subject by weighting "strongly hyperarticulated" utterances as 1, "somewhat hyperarticulated" utterances as 0.5, and "nonhyperarticulated" utterances as 0, and taking the mean weight across all utterances in the scenario. The 12 subjects who heard the instructions (the "instruction group") had lower mean hyperarticulation scores, 0.22 as compared with 0.60 for the 12 subjects who received no special instructions (the "no instruction group"), a significant difference $F(1,22) = 11.97$, $p < .01$.

Given that the instruction group had significantly fewer hyperarticulated utterances, and given that hyperarticulation is associated with lower recognition accuracy, we would expect the instruction group to have better recognition performance overall. However, although the trend was in that direction (18.1% word error for the instruction group versus 22.5% for the no-instruction group), the difference was not reliable. One possible explanation is a lack of power in the analysis, as a result of the small number of subjects and large individual differences in error rates. A second, not necessarily conflicting explanation is that the subjects given the instructions to "speak naturally" used somewhat less planned and less formal speech. We noticed that these subjects tended to have more spontaneous speech effects, such as verbal deletions, word fragments, lengthenings and filled pauses. Overall, spontaneous speech effects occurred in 15% of the 232 utterances for the instruction group, compared with 10% for the 229 utterances for the no-instruction group. Although these baseline rates are low, they may nevertheless have contributed to poorer recognition rates (see Butzberger et al. [2]). They may also be indicative of subtle speech style differences between the two groups not captured by the coding of hyperarticulation.

4. CONCLUSION

Application development can benefit from analyses of factors affecting system performance and user satisfaction. We have presented examples of ways in which the behavior and satisfaction of subjects interacting with an SLS may be affected. We have described ways in which parameters of the system itself, such as speed and accuracy, affect different aspects of user satisfaction. We have examined the effect of user experience on recognition performance and found a decrease in word error rate over repeated scenarios. Adaptation was relatively greater for those subjects who had more than 20% errors on the first scenario. The decrease in errors could be attributed at least in part to a decrease in sentence perplexity and to a reduction in the use of out-of-vocabulary words. We have also shown a significant relationship between word error rates and hyperarticulation, a speech style that occurs relatively frequently with an imperfect recognizer. We have shown that instructions not to hyperarticulate reduced this maladaptive speech style, but that instructions did not result in improved recognition performance overall.

Our studies have shown that along some dimensions, humans are flexible and can adapt in ways that improve system performance. However, hyperarticulation may be a maladaptive behavior for which a technological solution should be investigated. In particular we have found that strategies people use to try to improve normal human communication (e.g., hyperarticulation) can have the reverse effect in the context of our current models. While hyperarticulation is an "exaggerated" speech style that might improve comprehension for humans, it can cause poor recognition for automatic systems in which "exaggeration" is not adequately modeled.

Acknowledgments

We gratefully acknowledge support for this work from DARPA through the Office of Naval Research contract N00014-90-C-0085. The government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the government funding agencies. We also gratefully acknowledge Steven Tepper for software development.

REFERENCES

1. Bly, B., P. Price, S. Tepper, E. Jackson, and V. Abrash, "Designing the Human Machine Interface in the ATIS Domain," *Proc. Third DARPA Speech and Language Workshop*, pp. 136-140, Hidden Valley, PA, June 1990.

2. Butzberger, J. W., H. Murveit, E. Shriberg, P. Price, "Spontaneous Effects in Large Vocabulary Speech Recognition Applications," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

3. Ferguson, C. "Towards a Characterization of English Foreigner Talk," *Anthropological Linguistics*, 17, pp 1-14, 1975.

4. Jackson, E., D. Appelt, J. Bear, R. Moore, A. Podlozny, "A Template Matcher for Robust NL Interpretation," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

5. Kowtko, J. C. and P. J. Price, "Data Collection and Analysis in the Air Travel Planning Domain," *Proc. Second DARPA Speech and Language Workshop*, pp. 119-125, Harwichport, MA, October 1989.

6. MADCOW, "Multi-Site Data Collection for a Spoken Language System," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

7. Murveit, H., J. Butzberger, and M. Weintraub,"Speech Recognition in SRI's Resource Management and ATIS Systems," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

8. Murveit, H. and M. Weintraub, "Real-Time Speech Recognition System," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

9. Murveit, H., J. Butzberger, and M. Weintraub, "Performance of SRI's Decipher Speech Recognition System on DARPA's ATIS Task," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

10. Price P., "Evaluation of Spoken Language Systems: The ATIS Domain," *Proc. Third DARPA Speech and Language Workshop*, pp. 91-95, Hidden Valley, PA , June 1990.

11. Stern, R. M. and A. I. Rudnicky, "Spoken-Language Workstations in the Office Environment," *Proc. Speech Tech' 90*, Media Dimensions, 1990.

12. Weintraub, M., G. Chen, P. Mankoski, H. Murveit, A. Stolzle, S. Narayanaswamy, R. Yu, B. Richards, M. Srivastava, J. Rabay, R. Broderson, "The SRI/UCB Real-Time Speech Recognition System," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

USER BEHAVIORS AFFECTING SPEECH RECOGNITION

Elizabeth Wade, Elizabeth Shriberg, Patti Price

SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025 USA

ABSTRACT

We attempt to explain a decrease in recognition word error rate observed when users interacted over time with a spoken language system. We found no change in the language used (as measured by sentence perplexity), and only a small decrease in the number of out-of-vocabulary words. However, a behavior adversely affecting recognition, hyperarticulation, decreased over time. In addition, the acoustic match of hyperarticulated utterances to the system models also improved over time. We conclude that improvement in recognition was due to changes in speech rather than in language.

I. INTRODUCTION

Changes in the way users speak as they interact with a spoken language system over time may have consequences for recognition performance. Because humans are highly adaptive, initial recognition performance may not accurately predict later performance. System developers can benefit from considering not only initial use of a system, but also experience of a user over time. In addition, speakers interacting with a spoken language system may not exhibit the same language behavior observed in training data. Earlier, we found that recognition errors decreased as subjects interacted with the system over time [1]; the current paper more closely examines the source of this error reduction by looking at both the language and speech style of users.

Analyses were based on data collected using SRI's spoken language system (SLS), as part of a multisite collection effort [2] in which subjects solved air-travel planning scenarios. The SRI SLS combines the DECIPHER™ recognizer [3] with a robust natural-language understanding component [4], implemented in the air-travel planning domain. The system does not prompt the user for specific input; it simply accepts user-formulated queries. For example, the user might ask, "Show me flights from San Francisco to Philadelphia during the morning," to which the system should respond by displaying a table of flight information fitting those specifications.

In a previous paper [1] we reported that subject's word error rates decreased from Scenario 1 to Scenario 2. In that analysis we attempted to explain the source of this decrease; however, the addition of data in the current paper allows us to explain the phenomenon in further detail. We examine two potential causes for the decrease in error: changes in language and changes in speech style.

One possible explanation for the decrease in error is that users were changing their language to use more constructions of the types most easily recognized by the system. To test this hypothesis, we compared the perplexity of sentences in Scenarios 1 and 2 for each subject. If perplexity (essentially a measure of how unexpected a word sequence is given the system models) decreased in Scenario 2, we could conclude that subjects' behavior changed in a way that adapted to the language models of the system.

A second not contradictory hypothesis is that subjects were changing their speech style over time to better match the system's acoustic models. We coded and measured one speech style, hyperarticulation, which we had reason to believe would lead to recognition errors. If hyperarticulation was related to errors, and if the frequency of hyperarticulation decreased in Scenario 2, we could conclude that subjects' behavior changed in a way that adapted to the acoustic models of the system.

II. METHOD

2.1. Subjects

We collected speech and session logs for two scenarios from each of 24 subjects, counterbalancing for the selection and order of the scenarios they solved. The majority of subjects (17) were SRI employees recruited from an advertisement in an internal newsletter; a small number were students from a nearby university or members of a volunteer organization. Subjects were native speakers of English, ranged in age from 22 to 71, and had varying degrees of experience with travel planning and computers.

2.2. Materials

Four different travel-planning scenarios were used. One involved arranging flights to two cities in three days; a second involved finding two fares for the price of a first class fare; a third required coordinating the arrival times of three flights from different cities; and a fourth involved weighing factors such as fares and meals in order to choose between two flight times. Because the task demands of the scenarios were different, we controlled for scenario in the analyses.

2.3. Apparatus

The data were collected using SRI's Spoken Language System with no human in the loop. The basic characteristics of the DECIPHER™ speech recognition component are described in Murveit et al. [5,6], and the basic characteristics of the natural language understanding component are described in Jackson et al. [4]. The subjects used the real-time hardware version of the DECIPHER™ system which had a vocabulary size of 1,250 words [3,7].

SRI's SLS technology was implemented in the air travel planning domain, with which many people are familiar (see Price, [8]). The underlying database was a relational version of an 11-city subset of the *Official Airline Guide*. Recognition was based on the input of a Sennheiser HMD close-talking microphone.

The interface presented the user with a screen showing a button labeled "Click Here to Talk." A mouse click in this box caused the system to capture speech starting a 1/2 second before the click; the system automatically determined when

the speaker finished speaking based on silence duration set at a threshold of 2 seconds. Once the speech was processed, the screen displayed the words recognized, a "paraphrase" of the system's understanding of the request, and, where appropriate, a formatted table of data containing the answer to the query. When the natural-language component could not arrive at a reasonable answer, a message window appeared displaying one of a small number of error messages. A log file was automatically created, containing time stamps marking each action by the user and by the system.

2.4. Procedure

Subjects were seated in a quiet room and were given a short demonstration on how to use the system. Half of the subjects were given additional instructions explaining that, while they might have a tendency to enunciate more clearly in the face of recognition errors, they should try to speak naturally, since the system was not trained on overenunciated or separated speech. Once subjects were comfortable with the system, they were left alone in the room to solve the scenarios.

III. ADAPTATION

We compared Scenarios 1 and 2 for each subject to determine whether there were any changes in user behavior over time. Although subjects were not told to solve the scenarios as quickly as possible, they nevertheless took less time (10.5 compared to 13.0 minutes) to complete the second scenarios, $F(1,23) = 5.78$, $p < .05$. This difference was partially attributable to a lower number of total utterances in Scenario 2. In addition, we found significantly lower recognition error rates in subjects' second scenario. The mean word error rate was 20.4% for Scenario 1, but fell to 16.1% for Scenario 2, $F(1,22) = 5.60$, $p < .05$.

3.1. Language

We first hypothesized that this change in error rates might be due in part to adaptation to the language model of the recognizer. As a measure of deviation from the system's bigram language models, we used testset perplexity, which was based on the bigram probabilities of the observed word sequences. Perplexity measures the average likelihood (according to the system's models) that each word in a user's query will be followed by the next word, taking into account the base rate frequencies of the words. So a commonly phrased query like "I'd like to fly from San Francisco to Philadelphia" would have a low perplexity, since the system models would predict that each word is quite likely to follow the word that precedes it.

We confirmed the relationship between perplexity and word error in our data; there was a significant, positive average correlation between utterance word error and utterance perplexity, mean $r = .28$, $t = 4.55$, $p < .001$. Thus one way for subjects to improve recognition accuracy would be to change their language to conform to the language models of the system. For example, subjects might alter their initial language to use more common, easily recognized word sequences and to avoid rarer sequences that might tend to have more errors. However, we did not find support for this hypothesis. Perplexity decreased only slightly from 1 to 2 Scenario, with a geometric mean of 17.7 and 16.9, respectively. The magnitude of this difference was not significant given the variability both within and across subjects; perplexity within a scenario ranged from 8.8 to 38.9. The difference was nonsignificant by a Sign test, $p > .50$.

In an attempt to find converging evidence that changes in perplexity did not cause the decreased error rates, we obtained recognition results for the same sound files using a software version of the recognizer and two types of models. The bigram models were essentially the same as the original models used by the hardware, and were used as a control. The nogram models used only acoustic and word frequency information for recognition; they did not reflect any information about cross-word probabilities. Thus the recognition results from nogram models would not be improved by any user adaptation to the grammar of the original recognizer, whereas the results from the bigram models would. If the bigram results showed a greater decrease in word error than the nogram results, we could conclude that some of the decrease was due to adaptation to the language models. Figure 1 shows the word error recognition results from the two types of models. Both bigram and nogram results show essentially the same decreasing slope. This again suggests that adaptation to the language models was not a major cause of improved recognition over time.

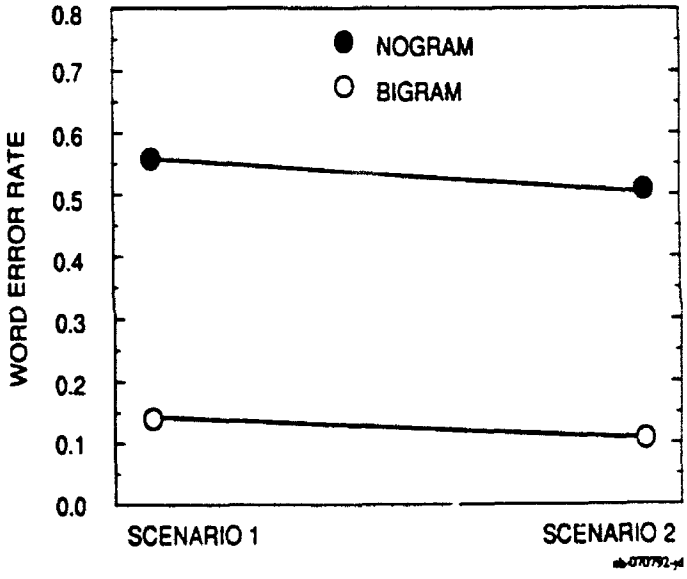


Fig. 1 - Bigram and nogram word error rates over time

We also examined whether the subjects tended to reduce their use of out-of-vocabulary words in Scenario 2. Subjects averaged 1.2 (less than 0.01%) out-of-vocabulary words in Scenario 1, as compared with 0.5 (also less than 0.01%) in Scenario 2. The number of these occurrences is so small as to be trivial; furthermore, the trend is nonsignificant, $F(1,21) = 1.74$, $p > .10$. This suggests that the use of fewer out-of-vocabulary words had little if any effect on overall recognition rates.

3.2. Speech Style

Having found no evidence for adaptation to the language models, we concluded that recognition improvement must be due to changes in user speech style. That is, as speakers became more familiar with the system, they learned to speak in ways that better matched the acoustics of the training data.

In human-human interaction, when an addressee (such as a foreigner) has difficulty understanding, speakers change their speech style to enunciate more clearly than usual [9]. We predicted that a similar effect might occur for people speaking to a machine with less than perfect understanding. We noticed

that, when using an SLS as opposed to a wizard-mediated system [10], subjects tended to hyperarticulate: releasing stops, emphasizing initial word segments, pausing between words, and increasing vocal effort. Since most of the data used to train the DECIPHER™ recognizer came from wizard-mediated data collection, where recognition performance was nearly perfect, examples of "frustrated" speech were rare. For this reason, we predicted that hyperarticulation would impair recognition performance, and that perhaps the lower error rates in Scenario 2 might be due to a decrease in the frequency of hyperarticulation.

Although hyperarticulation is a multifaceted behavior, it was nevertheless possible to make global judgments about individual utterances. Hyperarticulation was coded for each utterance on a three-point scale by listening to the utterances. Utterances were coded as (1) clearly natural-sounding, (2) hyperarticulated in portions, or (3) hyperarticulated throughout the utterance. The coding was done blindly without reference to session context or recognition outcome.

Using a within-subjects design, so that any differences in recognition performance could be attributed to a change in speech style, rather than speaker effects, we analyzed the speech style for Scenarios 1 and 2 of the same 24 subjects. Because not enough speakers had utterances in all three categories, we combined the hyperarticulation coding of two levels for statistical purposes. For the 21 subjects who had both natural and hyperarticulate utterances, we compared recognition performance within subjects and found that the hyperarticulate utterances resulted in substantially higher word error rates, 0.25 as compared with 0.11, $F(1,20) = 15.68$, $p < .001$.

Given that hyperarticulation leads to more errors, it is possible that the overall decrease in error rates is due to a decrease in the rate of hyperarticulation. In fact, the frequency of hyperarticulated utterances decreased from an average of 46% of utterances to 30% from 1 to Scenario 2, $F(1,23) = 4.97$, $p < .05$. The decrease was more pronounced for the completely hyperarticulate utterances than for the partially hyperarticulate. As shown in Figure 2, users tended to use proportionally fewer completely hyperarticulate utterances in Scenario 2. Since this indicates a trend toward fewer hyperarticulated words within utterances, this finding may also help explain the decrease in error rate.

Converging evidence for the effect of frequency of hyperarticulation on overall recognition rates came from the experimental manipulation of instructions. Of our 24 subjects, 12 had been given instructions not to "overenunciate." Under these instructions, subjects hyperarticulated less, on 4.3 or 28.0% of all utterances as compared with 7.5 or 52.5%. This effect was reliable, $F(1,22) = 5.00$, $p < .05$. Since hyperarticulation rates decreased with instructions, we expected a comparable decrease in error rates. We compared word error rates for the two instruction groups and found that the subjects who received the instructions tended to have lower word error rates overall 0.15, as compared with 0.20; however, this effect was not significant. As Figure 3 shows, there was no interaction between instructions and session; both the instruction and no-instruction groups had similar decreases in word error rate over time. Figure 3 also shows the comparable rates of hyperarticulation. As with error rate, there was no significant interaction between instructions and hyperarticulation rate. Thus a decrease in hyperarticulation was associated with a decrease in word error both when observed over time and when manipulated by subject instructions. This finding suggests that a decrease in the rate of hyperarticulation may account for some or much of the decrease in error over time.

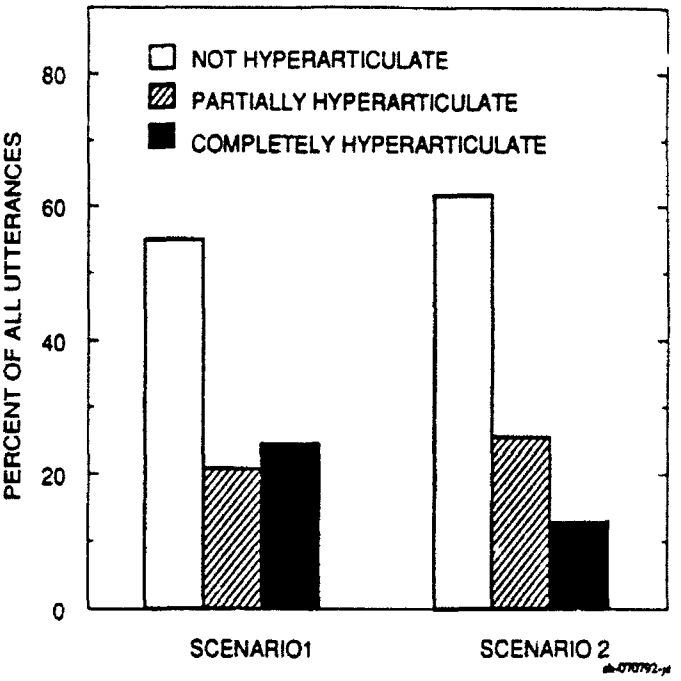


Fig. 2 - Frequency of hyperarticulate utterances over time

In addition to frequency of hyperarticulation, it is possible that the nature or degree of hyperarticulation may have changed over time. If hyperarticulated utterances themselves became more like the training data over time, this improved match might also have contributed to the reduction in error rate. Nonhyperarticulated utterances might also have become more similar to the training data. We measured the acoustic match between the utterances and the training data by running a forced alignment recognizer on the recorded sentences. Hidden Markov models associated with the sentence transcriptions were aligned to the VQ sequence produced by each sentence sound file. This procedure obtained the probability of each sentence's VQ sequence given the hidden Markov models it was aligned to.

Figure 4 shows log probabilities for hyperarticulated and nonhyperarticulated utterances in both scenarios. While the acoustic match for nonhyperarticulated utterances did not change over time, the match for hyperarticulated utterances improved sharply from the Scenario1 to Scenario 2. Because few subjects had utterances in all four categories (both hyperarticulated and nonhyperarticulated, in both scenarios), statistical tests were inappropriate. However, we observed a similar improvement in acoustic match for hyperarticulated utterances for both instruction groups, suggesting that the trend is not random. Thus, an additional factor contributing to lower recognition error rates is a change in the acoustic nature of hyperarticulated utterances over time

IV. CONCLUSION

We found that recognition word error rates decreased as users interacted with the same SLS over time. We found that this effect was due to changes in speech style rather than in adaptation to the language models of the system. We conclude that changes in one speech style, hyperarticulation, affect recognition rates in two ways. Users both decrease the rate of

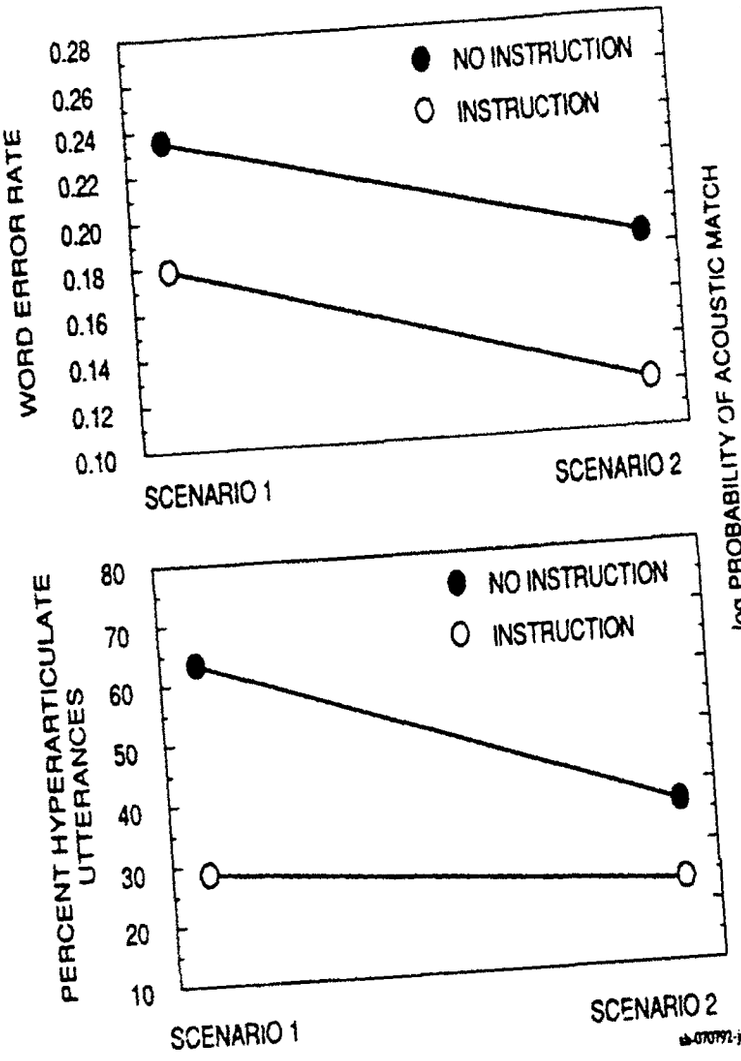


Fig. 3 - Effect of instructions to avoid overenunciation on word error and hyperarticulation rate over time

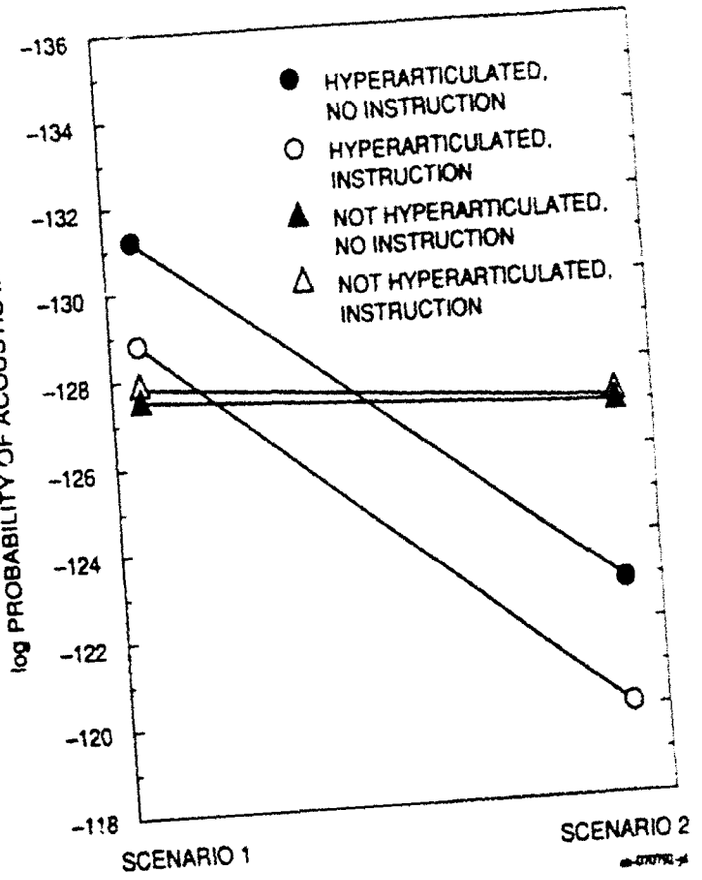


Fig. 4 - Deviation from perfect acoustic match between utterances and system models

hyperarticulation and alter the way in which they hyperarticulate so as to better match the system's acoustic models. Together, these two adaptations may account for the improvement in recognition rates observed as subjects use the system over time.

ACKNOWLEDGMENTS

We gratefully acknowledge the work of Steven Tepper for system design and development, and John W. Butzberger for assistance and analyses. This research was supported by the Defense Advanced Research Projects Agency under Contract ONR N00014-90-C-0085.

REFERENCES

[1] Shriberg, E., E. Wade, P. Price, "Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992

[2] MADCOW, "Multi-Site Data Collection for a Spoken Language Corpus," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992

[3] Murveit, H. and M. Weintraub, "Real-Time Speech Recognition System," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

[4] Jackson, E., D. Appelt, J. Bear, R. Moore, A. Podlozny, "A Template Matcher for Robust NL Interpretation," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

[5] Murveit, H., J. Butzberger, and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS Systems," *Proc. DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

[6] Murveit, H., J. Butzberger, and M. Weintraub, "Performance of SRI's Decipher Speech Recognition System on DARPA's ATIS Task," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

[7] Weintraub, M., G. Chen, P. Mankoski, H. Murveit, A. Stolzle, S. Narayanaswamy, R. Yu, B. Richards, M. Srivastava, J. Rabay, R. Broderson, "The SRI/UCB Real-Time Speech Recognition System," *Proc. DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992.

[8] Price P., "Evaluation of Spoken Language Systems: The ATIS Domain," *Proc. DARPA Speech and Language Workshop*, Morgan Kaufmann, 1990.

[9] Ferguson, C. "Towards a Characterization of English Foreigner Talk," *Anthropological Linguistics*, vol. 17, pp. 1-14, 1975.

[10] Sly, B., P. Price, S. Tepper, E. Jackson, and V. Abrash, "Designing the Human Machine Interface in the ATIS Domain," *Proc. DARPA Speech and Language Workshop*, Morgan Kaufmann, 1990.